Fast Identification Algorithms for Manipulating Biological Cells

A Thesis

Submitted to the College of Graduate Studies and Research

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in the

Department of Electrical Engineering

University of Saskatchewan

Saskatoon, SK S7N 5A9, Canada

By

Blessing Ohwo Diejomaoh

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical Engineering

University of Saskatchewan

Saskatoon, Saskatchewan, Canada S7N 5A9

Abstract

Fast Identification Algorithms and their Implementation for Physical Manipulation of Biological Cells

By
Blessing Ohwo Diejomaoh

The physical manipulation of biological cells is very attractive now in biotechnology (Butler, 1991)) because it opens the possibility of examining and manipulating single molecules. Other methods are based on chemical effects, electrical effects, etc., and they generally do not allow researchers to examine single molecules cell and, thus, to understand their interaction which may encode many useful pieces of information. Such physical manipulation is fully performed by robotic devices.

In order to automate the process of physical manipulation, micro machine vision for the fast identification of the objects involved is required. Typical objects that are involved are cells, cell elements, holders and injectors.

In the research described in this thesis, which was carried out in the Advanced Engineering Design Laboratory of the Mechanical Engineering Department, University of Saskatchewan, algorithms for the three objects (the cell, holder and injector) were developed, implemented and tested. The results obtained have shown that the fastest identification times for these three objects are respectively 0.12s for the cell oocyte,

6.78s/100 frames for the holder, and 6.72s/100 frames for the injector. These performances are acceptable in the context of the physical manipulation of biological cells

The goal of the research described in this thesis was to develop algorithms that would give a fast recognition of the cell manipulation system. With the aid of the algorithms, an automatic operation of the cell manipulation system would be achieved. Image process and pattern recognition techniques were used in developing the Visual C++® GUI algorithms that would automatically recognize the components of the cell manipulation system for the purpose of manipulating the cells.

Acknowledgements

I would like to express my profound gratitude to Dr. Chris Zhang and Dr. Madan Gupta for the supervision and advice that they provided during my graduate studies and the preparation of this thesis. Also, I appreciate the assistance of Mr. Puren Ouyang and Ms. Elizabeth Nikiforuk.

My gratitude goes to my parents: Pa. J.A.E Diejomaoh (blessed memory), Mrs. Regina Diejomaoh and all my siblings. Also, I am grateful to all my friends, relatives and teachers too numerous to mention, who contributed in various ways during my educational pursuit. However, worthy of mention are my high school teacher Mrs Gupinathan and my friends Viano Oghenekevwe and Soji Oyenuga.

Finally, and most importantly, I am grateful to God almighty for sustaining me, without whom, I would not have been able to complete this thesis successfully.

Dedication

I dedicate this thesis to my late father Johnson Diejomaoh and late sister Edith Diejomaoh. I miss them dearly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

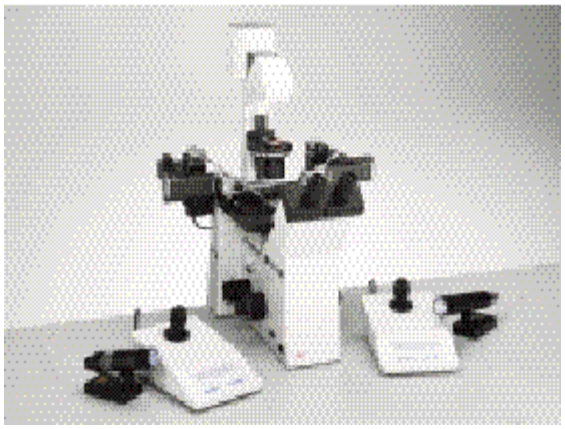
1.1 Biological Cell Manipulation Automation

Biotechnology has given rise to a tremendous breakthrough in medicine and agriculture. Biotechnology uses molecular biology and genetics to create new agricultural products, foods, better industrial materials and medicines. Millions of people have been helped or healed by drugs and vaccines developed with the help of biotechnology. Today, biotechnology is improving crop yields, while conserving water, soil and the environment. In the very near future, biotechnology may provide consumers safe foods with enhanced nutritional quality, while offering additional environmental benefits. In a long line of scientific advances, biotechnology represents the next step forward, allowing researchers to incorporate into plants one or many beneficial traits from a wider variety of sources. According to David Betsch of the Biotechnological Training Programs (BTP), IOWA State University, a narrower and more specific definition of biotechnology is "the commercial application of living organisms or their products, which involves the deliberate manipulation of their DNA molecules"

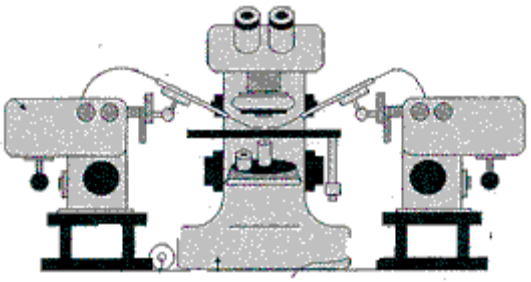
Biotechnology involves a wide range of biological cell operations (Butler, 1991). These biological cell operations, implemented in genetic engineering, include cell injection, cell cutting, cell fusing and cell holding. Cell injection involves injecting a female or egg cell with the genes of a male or sperm cell. The fusion of the two, male and female genes, occurs in the nucleus. The cells can also be cut or fused together during the

process of cell manipulation. The biological process of holding the cell in position is carried out by the holder.

Conventionally, cell injection has been conducted manually. TransferMan® NK micromanipulator is an example of a manually operated cell manipulator. It is an adaptation of the common Leitz model. The diagram of the PM20 is shown in Figure 1.1



a



b

Figure 1.1: Examples of a manually operated cell manipulations system.

Operators need at least a year of full time training and success rates are disappointingly low and inaccuracy is very high. One reason for this is that successful injections are not precisely reproducible.

A successful cell injection rate is determined by the control of force and speed. Operators currently depend more on their intuitive feelings than on quantitative measurements such as how much force and what speed should be applied. This occurs due to the fact that there are few quantitative measurements available in the cell injection process. With quantitative measurements such as force and speed (Zhou et al., 1998), automated cell injection can be highly reproducible. The second reason for the low success rate of the conventional cell injections is due to contamination issues, which calls for the elimination of human involvement. Thus, automated cell injection is desirable since it is training-free, contamination-free and is highly reproducible, greatly increasing the success rate (Advance Microsystem Laboratory, University of Minnesota).

Also, it is necessary to insert a micropipette into the cytoplasm in case of sperm injection or DNA transplantation (Hogan et al., 1994). By using conventional hydraulic manipulators, it has been found that the cell undergoes large deformation due to the cell membrane elasticity. This deformation can even destroy the nucleus. Therefore, a rapid and smooth injection process is required and this can be achieved using a piezo impact drive mechanism. For a fully automated system, image processing is applied to identify the cell oocyte.

The oocyte is a gametocyte that develops into an ovum after two meiotic divitions. The ability to accurately identify the oocyte where the nucleus resides is very important in injecting genes into the nucleus, since most of the genetic transformations start in the nucleus of the cell. Also, identification of the oocyte of the cell will assist the left hand of the cell manipulation system in accurately locating the nucleus and injecting genes into it.

A number of systems have been developed for the purpose of which is to automating the operations on cells (Deok-Ho et al., 2001). These systems have much in common in their architecture. Figure 1.1 shows a system architecture being developed at AEDL, Advanced Engineering Design and Manufacturing Research Group, led by Dr. C. Zhang of the Department of Mechanical Engineering, University of Saskatchewan. The two manipulators, one for holding the cell (the left manipulator) and one for manipulating the needle (the right manipulator), are shown in Figure 1.2.

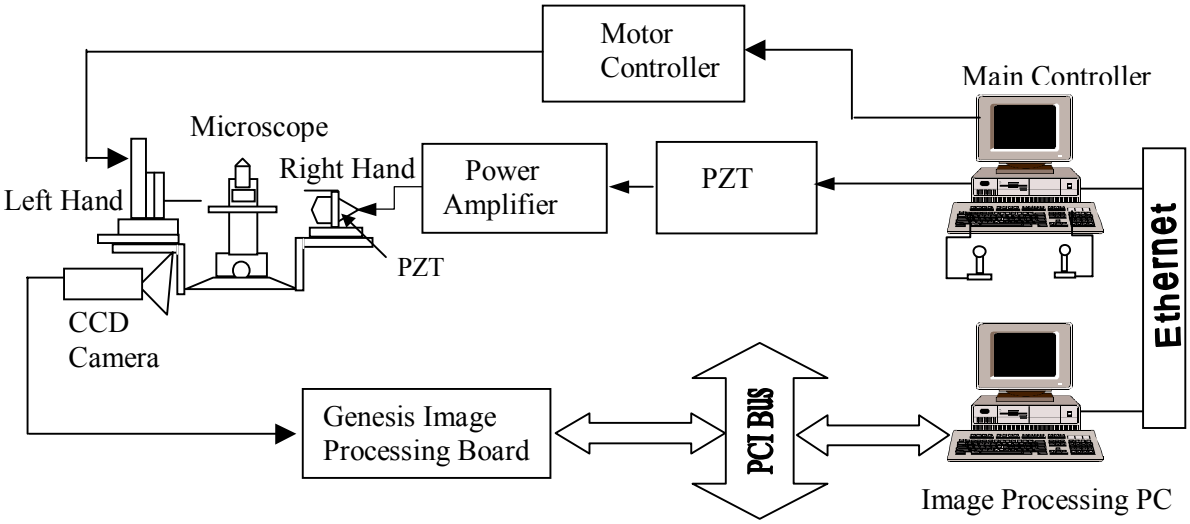


Figure 1.2: Block diagram of an automatic cell manipulation system.

In order to understand the image processing aspect of the automatic cell manipulation system, the system as a whole has to be understood. A cell manipulation system is a system that facilitates the remote handling of microscopic objects under the microscope, as seen in Figure 1.3.

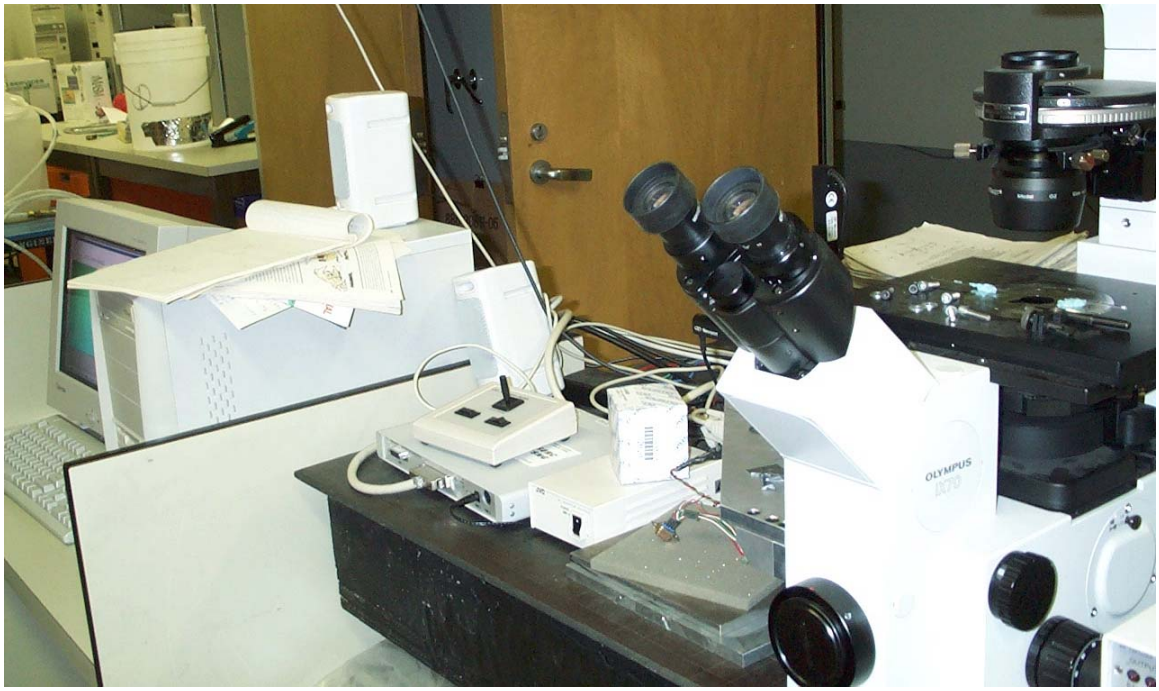


Figure 1.3: Automatic cell manipulation system.

The cell manipulation system consists of three modules: an executive module, a control module and the sensory module. The vision or image processing is part of the sensory module. The left and right manipulators form the executive module. The left manipulator is a 3-DOF (degree of freedom) XYZ stage with a workspace of $10x10x10mm^3$ and an accuracy of 1 micron.

The left hand of the cell manipulation system holds the cells in place, while the right hand does the manipulation of injecting, cutting and fusing (Cid-Arregui et al., 1997). The right hand of the CMS is a 3-DOF compliant mechanism. Piezoelectric actuators mounted on a compliant mechanism drive each link of the mechanism. The piezoelectric (PZT) controls the right hand of the cell manipulation system with strain gauges used as position sensors.

A micro injector is mounted on it to perform the gene injection. The diagram of the cell, the holder and the needle viewed under the microscope are shown in Figure 1.4.

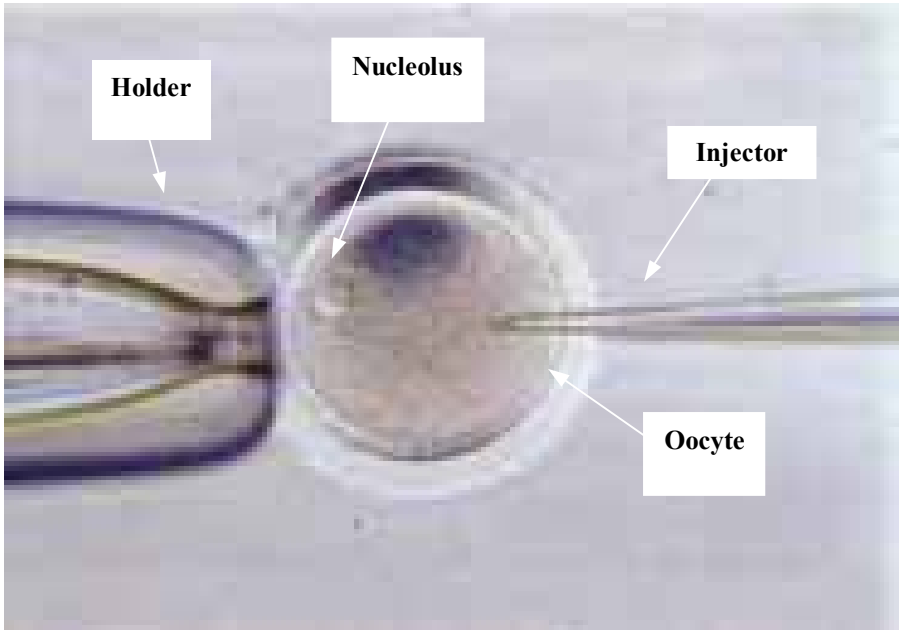


Figure 1.4: Image of the left hand (injector), right hand (holder) and cell.

(The courtesy of Higuchi, Torrii and Yamamoto Laboratory, University of Tokyo)

The micro vision system serves as a sensor for understanding the relationship between the three objects. The information about this relationship is then used by these controllers to instruct the actuators to drive the two manipulators to the target (that is, the nucleus or any part within a cell). This is the first feedback loop.

There is also a second feedback loop system in the manipulator system. This loop has a sensor, usually a strain gauge, at the joint actuator. The relationship between the end effector (for example the needle) and the joint actuator is known. Therefore, the control of the end effector (which is the ultimate goal) can be implemented through the inverse kinematics equation and the readings of sensors at the joint. The control block diagram is shown in Figure 1.5.

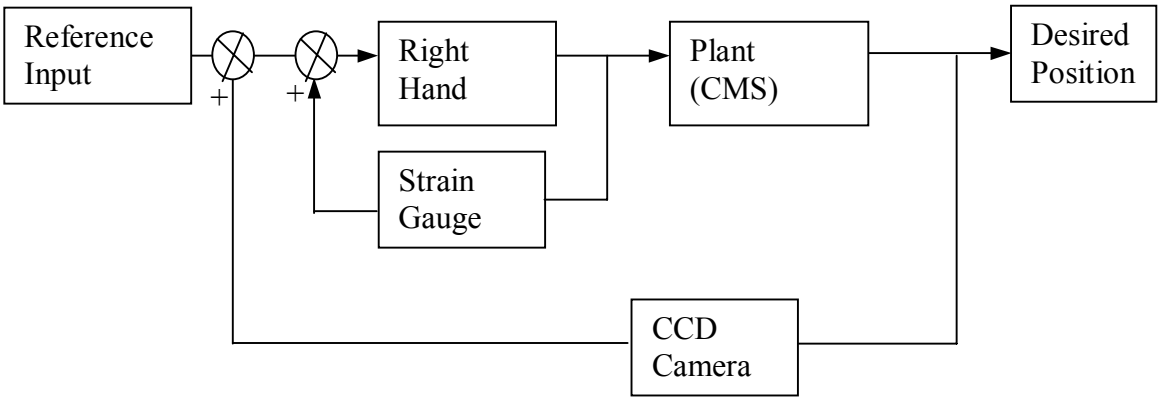


Figure 1.5: Block diagram of the feedback control system of the cell manipulator

1.2 Motivation

Information about the objects using a light microscope with a CCD camera is obtained through imaging process and analysis. The rate of sending information to the actuators, from information capturing to decision generating, is dominated by the rate of the CCD camera and the rate of the image processing.

Typically, the rate of the CCD camera can be 100 frames per second. Thus, the time for making one frame (one image) is only 0.01 second, whereas the image processing analysis is usually a few seconds. Therefore, improving the rate of image processing analysis is a key issue in increasing the control speed of the micromanipulator system.

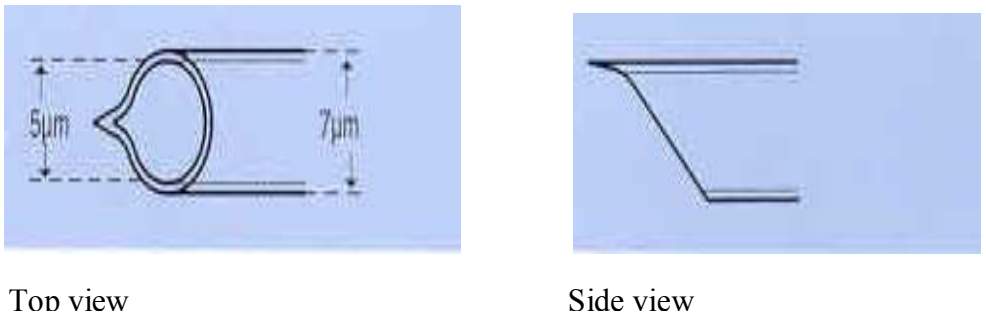
The research described in this thesis addresses the above important issue. That is, the goal of this research was to develop a fast image analysis method in the context of biological cells injection. The research considered three objects involved in these types of cells operations; namely the injector, the cell, and the holder.

The results and method of the research may also be used in machine vision (Papanikolopoulos, 1995). Real time control requires fast image processing and accurate identification of the objects. Therefore, the research carried out could be implemented in tracking the oocyte by the right hand of the CMS, during cell manipulation. An important component of the CMS or a robotic system is the acquisition, processing and interpretation of the available sensory information. Specifically, combining vision or image processing with control can result in better tracking of the oocyte of the cell (Sanderson et al., 1982). Better tracking and servoing correspond to more effective and more accurate active vision algorithms.

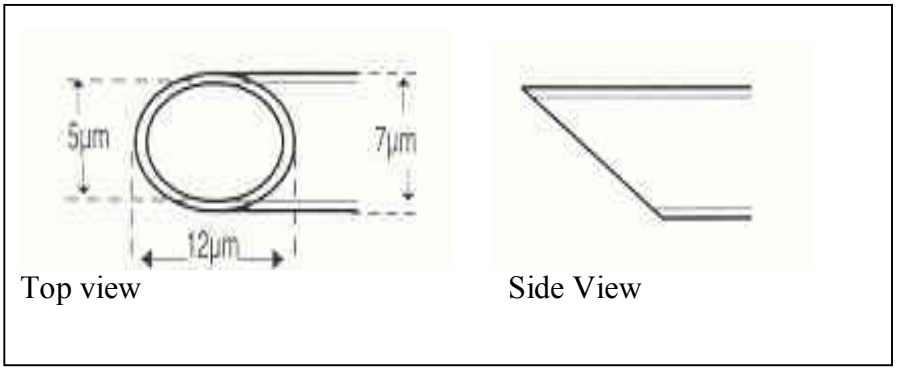
1.3 Related Work and General Feasibility

The injector is a hollow tube that is filled with genetic materials. As shown in Figure 1.6a, the injector has a long sharp end that is used to penetrate a cell. The geometry of the injector (Figure 1.6a) is illustrated in Figures 1.6b and 1.6c, respectively.

(a)



(b)

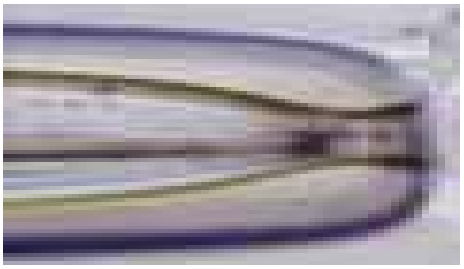


(c)

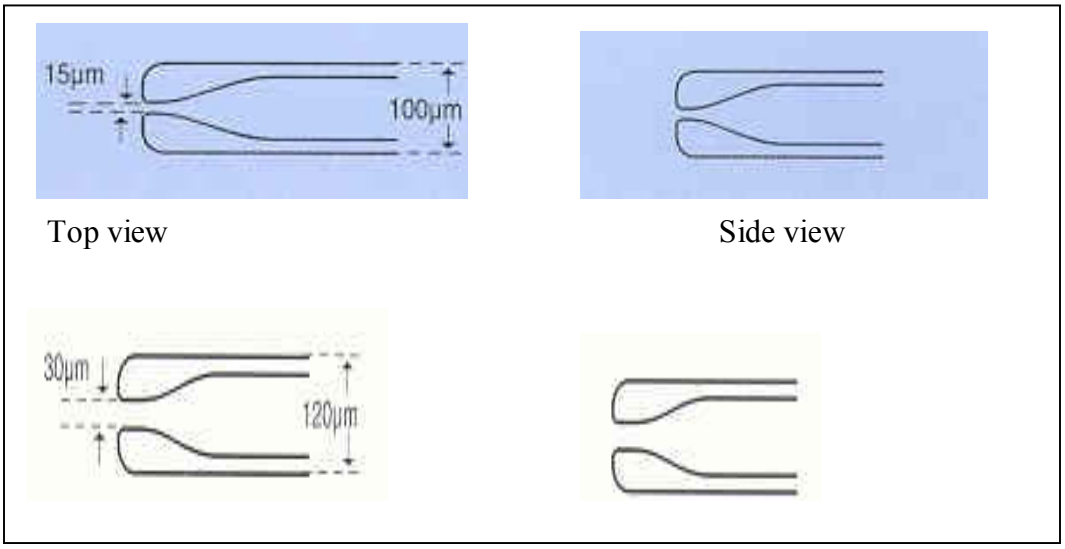
Figure 1.6: Injector.

(The courtesy of Research Instruments Ltd)

The holder is a hollow tube that holds the cell in place by suction. The holder has a fairly rectangular end with a wide surface area to serve the purpose of holding the cell. The geometry information of the holder is shown in Figure 1.7b.



(a)



(b)

Figure 1.7: Holder.

(Figures 1.7b is the courtesy of Research Instruments Ltd)

A typical biological cell considered in the present study is shown in Figure 1.8.

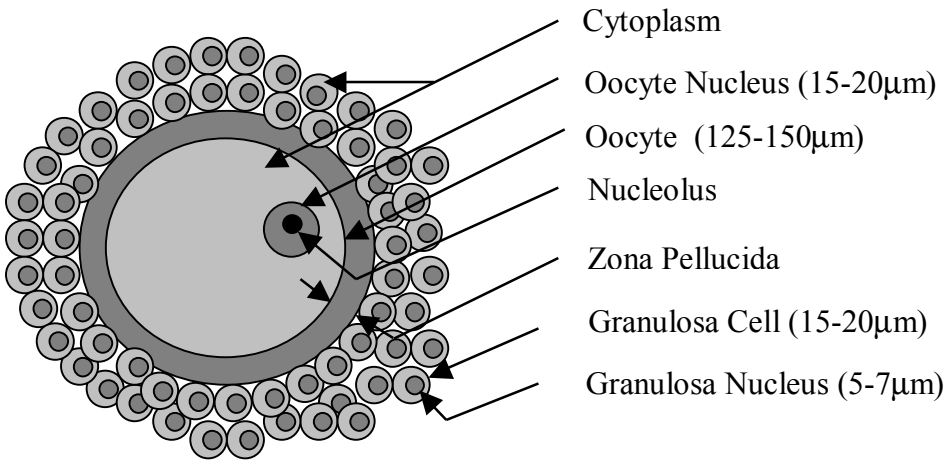


Figure 1.8: Anatomy of the Oocyte in a cell.

(The courtesy of Kirk Backstrom, University of Saskatchewan)

The key elements are the oocytes and nucleus. So far, no work has not been reported on using a light microscopy to identify the nucleus. The reason may be the limit of a light microscopy in terms of resolution.

For example, the resolution of a light microscopy is 0.825 micron/pixel. However, a manual cell manipulations system can usually identify where the nucleus is located with a very low efficiency and successful rate.

As far as the identification of the oocytes, reported studies (Xudong et al., 2001) have mainly followed two general approaches in image based identification of objects. They are

i) Feature matching

ii) Pattern matching

Feature matching methods require defining the features of an object that are unique with regards to its surroundings. These features, in the case of biological cells include contour and amplitude spectra. Feature extraction is a time consuming task (typically 0.11sec). Therefore, this approach is not suitable to real time feedback control applications. Pattern matching methods work directly on the image (Xudong et al., 2001).

1.4 Research Objective and Scope

The goal of this thesis was to develop a fast algorithm to track the three objects:

i) oocyte

ii) injector

iii) holder

The objectives of this study were defined as follows:

Objective 1: Develop a fast algorithm to track the oocyte/cell nucleus,

Objective 2: Develop a fast algorithm to track the injector,

Objective 3: Develop a fast algorithm to track the holder.

The implementation of the algorithm was carried out by using a general-purpose image processing package, the Matrox® Imaging Library (MIL).

1.5 General Research Methods

An investigation was carried out on the previous recognition algorithm that were not solely based on recognizing the cell nucleus (Gonzalez et al., 1992), with a view of developing the method for the algorithm used in this research. This was carried out in the form of investigating some of the research literature.

Then, the software language for implementing the algorithm was decided upon. Visual C++® was chosen because of its versatility and the Graphic User Interface (GUI) capabilities it possesses.

The image feature extraction algorithm was carried out by the image-processing computer connected to a camera fixed to the workspace; that is, the workspace of the microscope. The algorithm developed entailed using the Visual C++® program written in a Matrox imaging software platform. The Visual C++® program was developed by calling the functions contained in the Matrox image processing library. The functions, when called, executed the feature extraction algorithm. The Visual C++® algorithm has a GUI interface that serves as an interface between the user and the PC. Digital images of the cells were obtained via the CCD camera and sent to the GUI interface of the image processing PC. These digital images were further processed to identify the cell nucleus. The algorithm was tested on both the single cell and multiple cell digital images. The position of the cell nucleus of the single digital images was also determined during the run time of the code, or in the case of the multiple digital images, the positions of the cell nuclei.

In addition, a subroutine for comparing the speed of the approach with the speed of the pattern matching method was developed. To determine the time, the algorithm was executed for a number of iterations and the average time for one iteration was determined.

1.6 Outline of the Thesis

The background knowledge of this research, which includes the image processing theory and method, image processing software used, and object tracking, is presented in Chapter 2. Presented in Chapter 3 are various algorithms for the needle, oocyte and holder. Presented in Chapter 4 are the results of the experiment and discussions on the performance of the various algorithms, which lead to the optimal selection of the parameter in the algorithms. Chapter 5 contains the conclusions and includes a discussion of future work.

Chapter 2

Background Knowledge

2.1 Introduction

In this chapter, micro vision and image processing are briefly discussed which builds a context for the remainder of the discussion in the thesis. Basically, machine vision solutions for object tracking (Allen et al., 1991) have two classes of methodologies: the feature based approach and the pattern matching approach. The feature approach is discussed in Section 2.3. In Section 2.4 a discussion is given of the pattern matching approach.

2.2 Digital Image Definition

Digital processing converts a discrete input image into a discrete output image. To make this possible, the image, which requires processing, has to be converted from the continuous to the discrete form. A monochrome image is a two-dimensional light intensity function, *f(x,y)*, where *x* and *y* are the spatial coordinates and the value of *f* at *(x,y)* is proportional to the brightness of the image at that point. The brightness is also called the grey level. It is further possible to classify the brightness into three categories: the red, the blue and the green. For each of this classes there will be a function, *f(x,y)*.

Depending on the modalities used to acquire the image, the two-dimensional image plane can be divided into m by n grids. Then, the domain of the pixels is $\vartheta = \{(x, y), \cup 1 \le x \le x_m, 1 \le y \le y_n\}$. Where x_m and y_m are the maximum image co-ordinates. The range of the image function is {0,1,2…255} representing 255 grey levels.

2.3 Digital Image Processing

2.3.1 Image Contrast

Image contrast is related to the range of gray levels in an image; the greater the range, the greater the contrast and vice versa. The contrast C is defined in several ways. For example, $C_1 = GL_{max}/GL_{min,}$ $C_2 = GL_{max}-GL_{min}$, $C_3 = \sigma_{GL}$. In these expressions, $GL_{max,}$ GL_{min} are the maximum and minimum gray levels in the image and σ_{GL} is the gray level standard deviation.

2.3.2 Grey Level Manipulation

The examination of the image histogram is a step for manipulating the image contrast. The image histogram is the statistical distribution of the gray levels in terms of the number of pixels (or the percentage of the total number of pixels) having each gray level. The histogram for many images tends to be Gaussian in shape (Baldock et al., 2000)

2.3.3 Grey Level Morphology

Morphology entails probing the image with a structuring element. This is a set of pixels with a reference point. To perform morphological operations on an image of grey levels (rather than binary), the image is regarded as a height map; the brighter the pixel, the higher the surface (Horgan et al., 2000).

2.3.4 Type of Image Transformation Techniques

Image transformation attempts to replace the visual interpretation step with quantitative decision making. The output from a transformation processing is a thematic map in which each pixel in the original imagery has been classified into one class. The image transformation technique can be divided into two generic types: point processing and neighborhood processing.

Point processing consists of the transformation of each original image pixel value into a new value for the output image. Transformation depends on the grey level of the original image pixel. In other words, a point operation is one which transforms an input image *A(x,y)* into an output image *B(x,y)* in accordance with a law *f*, which may or may not be linear and is consistent with the whole image: B(x,y) = *f*[A(x,y)]. In the case of digitized images, each starting pixel is converted to another pixel at the same location, which derives its gray level from the original pixel the law *f*.

The neighborhood processing performs a transformation on each pixel in a way that depends not only on the gray level of the pixel being processed but also on the grey level of each of the pixels in the vicinity of the pixel being processed, and it includes such techniques as enhancement and interpolation.

2.3.5 Thresholding

Thresholding is a particularly useful technique for establishing boundaries in the images, like metaphases. It is computationally simple to implement and does not fail to define disjoint regions with closed, connected boundaries. When using a threshold rule, one selects a threshold grey level and then assigns all the pixels below that grey level to the background.

All pixels with grey level at or above the threshold are considered to be interior points. The boundary is then the set of interior points each of which has at least one neighbour outside the object. Thresholding works well if the objects of interest have reasonable uniform interior grey level and rest on a background of unequal but relatively uniform grey level. There are two types of thresholding methods: the fixed threshold and the adaptive threshold.

In fixed thresholding, the threshold grey level value is held constant throughout the image. Properly selected, a fixed global threshold grey level usually works well when the background level is reasonably constant throughout the image, and the objects all have approximately equal contrast with respect to the background.

In adaptive thresholding, the background level is not always constant and the object contrast varies within the image. In such cases, a threshold works well in one area of the image but might work poorly in another area. In these cases, it is convenient to use a threshold grey level that is a slowly varying function of position in the image.

2.3.6 Edge Detection

If the pixel falls on the boundary of an object, its neighborhood is a zone of grey level transition. Edge detection operators examine each pixel neighborhood and the direction of the grey level transition. If the edges are strong, and the noise level is low, one can threshold an edge magnitude image and thin the resulting binary image down to single-pixel wide closed connected boundaries.

2.3.7 Erosion and Dilation

The simple erosion process is the process of eliminating all the boundary points from an object, leaving it smaller in area by one pixel all around the perimeter. The diameter of a circular object decreases by two pixels with each erosion. Erosion is useful for removing from a segmented image objects that are too small to be of interest. Dilation is simply a process of incorporating into an object all the background points that touch the object, leaving it larger in area by that amount. If the object is circular, its diameter is increased by two pixels with each dilation. Dilation is useful for filling holes in segmented objects. Dilation, repeatedly applied, will merge all the objects into one.

Mathematically, the 3 by 3 common structuring element B (Xudong et al., 2001) is used whose form is as follows:

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

a) The erosion operation can be defined as the following operation:

Given a binary image ***J*** and a structuring element ***B***, the erosion of ***J*** relative to ***B*** is defined as:

$$J^e = ERODE(J, B)$$

$$J^e(i, j) = AND[B \cdot J(i, j)]$$

where *AND* denotes the logical or Boolean operation and J^e denotes the eroded binary image

$$AND(X_1, X_2, \cdots, X_{2M+1}) = \begin{cases} 1; & if \quad X_1 = X_2 = \cdots = X_{2M+1} = 1 \\ 0; & otherwise \end{cases}$$

Where X represent pixel values and X_{2m} represent pixel points

Erosion tends to decrease the sizes of objects in a binary image. Erosion is used to simplify the structure of an object. Objects or their parts with a small width will disappear. Erosion is therefore used to remove small particles in the image process.

b) The dilation operation can be defined as follows:

Given a binary image ***J*** and a structuring element ***B***, the dilation of ***J*** relative to ***B*** is defined as:

$$J^d = DILATE(J, B)$$

$$J^d(i, j) = OR[B \cdot J(i, j)]$$

where *OR* denotes the logical or Boolean operation J^d denotes the eroded binary image

$$OR(X_1, X_2, \cdots, X_{2M+1}) = \begin{cases} 0; & if \quad X_1 = X_2 = \cdots = X_{2M+1} = 0 \\ 1; & otherwise \end{cases}$$

Where X_i (i = 1,2…2_{M+1}) represents the grey level of a pixel say i

The process of erosion followed by dilation is called ‘opening’. It has the effect of eliminating small and thin objects, breaking objects at thin points and generally smoothing the boundaries of larger objects without significantly changing their area. The process of dilation followed by erosion is called ‘closing’. It has the effect of filling small and thin holes in objects. Often when noisy images are segmented by thresholding, the resulting boundaries are quite ragged, the objects have false holes and the background is

peppered with small noisy objects. The successive application of the opening or the closing operation can improve the image significantly.

2.3.8 Image Smoothing

Image smoothing is the set of local preprocessing methods whose predominant use is the suppression of image noise. Local image smoothing can effectively eliminate impulse noise or degradations appearing as thin stripes depending on a particular image software/hardware system used. For this purpose, a 3 by 3 neighborhood operator with the following convolution mask *h* is applied:

$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

2.4 Pattern Matching Approach

Pattern matching allows for the comparison, both visually and statistically, of two ratings from a concept map in order to explore consensus, track consistency over time or evaluate outcomes relative to expectations. The main function of the pattern matching module is to search for occurrences of a pattern in an image. The pattern being searched is called the search model and the image from which it is extracted is called the model's source image.

Young et al. (1995) employed template matching and edge detection to identify yeast cells. The template matching method proposed by Gonzalez and Woods (1992) locates matches of a sub-image with grey levels w(x, y) within an image of grey levels f(x, y), based on a "goodness of match" statistic evaluated at each pixel in the image.

The statistic here is the covariance, taking the value g(i, j) at the pixel location (i, j) within the image, where

$$g(i,j) = \sum_{k=-m}^{m} \sum_{i=-m}^{m} (f(i+k, j+1) - \overline{f}(i,j) * \left(w(k,l) - \overline{w}\right)$$

Where $\overline{w}$ is the template mean grey level and $\overline{f}(i, j)$ is a local image mean evaluated over the area being matched. High covariance values indicate a good match using a (2m+1) by (2m+1) template centered at (i, j).

A template was constructed by assigning grey levels to pixels within an ellipse (because the yeast cells were assumed to be ellipsoidal) according to their distances from the edge, the center pixel receiving an intensity of 1 and points at the edge and beyond an intensity of 0.

A 63 by 63 template, with the major axis length of 25 and minor axis of length of 15, was constructed using

$$f(i,j) = 1 - \left\{ \left(\frac{i}{\sigma_x} \right)^2 + \left(\frac{j}{\sigma_y} \right)^2 \right\}$$

and

$$w'(i,j) = w(i,j) - w(i-1, j-1)$$

where $\sigma_x = 25$, $\sigma_y = 15$ and if f(i, j)<0 then f(i, j) is set to 0. The grey levels were scaled to a range of (255,0).

For cell identification, two additional templates of sizes 49 and 39 pixels were constructed by applying a bilinear interpolation to the 63×63 templates (Glasbey and

Hogan, 1994). Three templates were chosen to correspond to the largest, middle-size and smallest cells in the image.

The method employed by Young et al. (1995) recognizes only the cells. Moreover, it is too cumbersome to implement in a cell manipulation system. It is fairly accurate, but computer intensive. For example, the CPU time taken to apply the largest template to the algae cell was 896 seconds on a SUN SPARC station 10.

2.5 Feature Matching Approach

The selection of a set of features which efficiently describe a system in terms of a pattern to be recognized in those features is itself pattern recognition. Each of the features describes some aspect of the pattern and amounts to a decomposition of the quality to be recognized in the overall problem into a set of easily recognized qualities.

This is illustrated in the figure below:

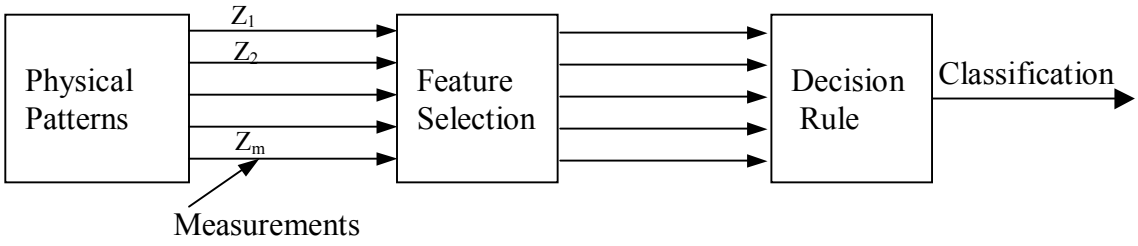


Figure 2.1: Block diagram indicating the use of features to classify physical patterns.

Feature selection requires a subset if n features are to be selected from a set of m possible control features f_i (i= 1…m), where m > n. Feature assignment addresses the

choice of which feature should be used to control the actuator. An examination of the features being measured should be made to ensure that the measurements make a reasonable representation of the required properties and that none of the measurements are inappropriate. In the case of chromosomes, there are two morphological features used by cytogeneticists in addition to the banding pattern: size and centromere position. Size could be measured as either length or area.

Graham (2000) employed features in classifying chromosomes. Morphological features of size and centrometric index were used to perform a partial classification into seven groups. The banding features, which are necessary to assign a chromosome to one of the 24 biological significant groups, are based on the density profile. There are many studies proposed (Carraza et al., 1998, Hogan et al., 1994). One early proposal (Glasbey et al., 1994) was to fit a series of Gaussian functions to the profile peaks and to use the parameters of the Gaussians (μ and σ) as classifier features.

The most significant features describing chromosome morphology are size and chromosome position. They are sufficient for it in dividing the chromosomes into seven classes, but not for identifying the homologous pair.

Chromosome size is reflected in the chromosome's length, area and integrated density. They are the features that can be normalized. The location of the centromere along the chromosome is valuable morphological information. In order to classify chromosomes one must find ways to qualify the unique characteristics of the band patterns. This requires knowledge of the chromosomere position. Once the chromosomes have been measured, the next step is to identify each one. Classification was carried out

in two steps: first, the probabilities of group membership were calculated, and then each chromosome was assigned to a position in the karyotype image based on those probabilities.

2.6 Summary

A review of existing studies on machine vision identification of cells was conducted in this chapter. Most of these studies focused on how to accurately identify macro molecules (e.g chromosomes) in the cells without taking into consideration the speed of identification. These studies are less useful to the intended work presented in this thesis. Some studies, among others (Lee and Wohn, 1988; Weiss et al., 1987; Yu and Nelson, 2003), were along the same line of this thesis work; yet they have not reported somewhat detailed information on machine vision identification of cells and molecules in cells with a special emphasis on the speed of identification.

Chapter 3

Algorithm and Implementation

3.1 Introduction

In line with the objective of this research, identifying the Oocyte of the cell, holder and injector, algorithms for identifying these components in the context of manipulation of the cells, are presented in this chapter. In the case of identifying the Oocyte, the algorithm was based on the feature matching technique because the Oocyte is distinct from other elements of its size. The pattern matching technique was used for identifying the holder and the needle, respectively. In Section 3.2, the algorithm for the Oocyte is elaborated. Two general approaches, feature matching and pattern matching, are presented for the purpose of further comparison. Section 3.3 gives the algorithm for the holder. In section 3.4, there is an elaboration of the algorithm for the injector. Section 3.5 gives the implementation of these algorithms and user-interfaces. Section 3.6 is a summary of this chapter.

3.2 Algorithm for the Oocyte

3.2.1 The Feature Matching Approach

The Oocyte is an element in which the nucleus resides. The Oocyte is usually present together with other elements such as mitochondria, vacuole and cytoplasm as shown in Figure 3.1. The objective of the algorithm is to identify the Oocyte out of these elements.

For this approach to work, the object under investigation must have features or combinations that are unique. Further, such unique features may possibly be identified using an imaging processing technique. In the case of the Oocyte, the size and color of it may be unique, or their combination may be unique. The size and color features are described below.

The size feature:

The oocyte is the largest cell particle in the cell cluster (see Figure 3.1) and is approximately circular. Its size is measured by the diameter of the particle.

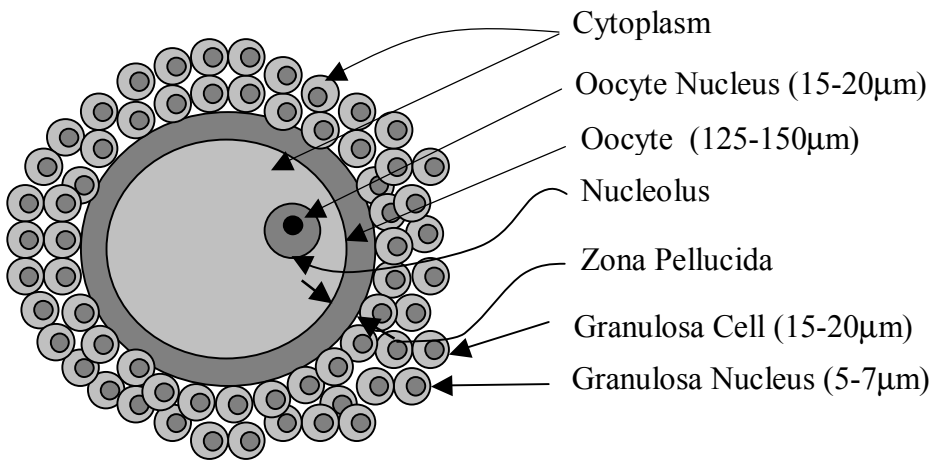


Figure 3.1: Anatomy of the Oocyte with granulosa cells.
(The courtesy of Kirk Backstrom, University of Saskatchewan)

The color feature:

The oocyte is darker than other elements and in most cases the Oocyte is the darkest particle among the cell cluster (see Figure 3.1). The algorithm was developed as follows.

In general, there are three processes to complete the task:

- Preprocess
- Main process
- Post process

(i) Preprocess

The process starts from the point where the image is located on the screen. The starting image is, in general, like the one shown in Figure 3.2 (Cid-Arregui et al., 1997). There are three objects:

- Cell clusters
- The holder
- Injector

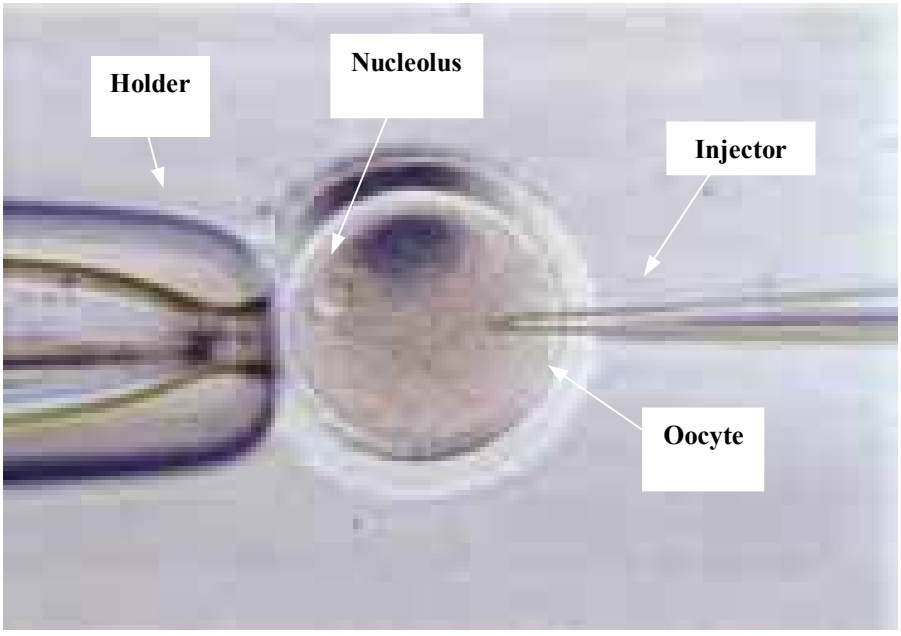


Figure 3.2: Starting image of the cell, injector and holder

(The courtesy of Higuchi, Torrii and Yamamoto Laboratory, University of Tokyo)

The cell cluster is the one shown in Figure 3.1. The preprocess is required to perform the following tasks with respect to the identification of the Oocyte in the cell cluster:

- Remove the noise on the image field.
- Remove the holder and the injector.

(ii) Main process

The main process completes the identification of the Oocyte in the cell cluster. The post conditions of this process are:

- The Oocyte is located (that is; its co-ordinates are found).
- Its perimeter, circumference or area is found.

In image processing, all the significant particles or elements in the image are regarded as a "blob" (a term used in image processing). The blob is thus defined as a set of connected pixels. The following are steps to complete the main process:

1) Locate all the blobs from the image. The post conditions of this operation are: (a) List of blobs with identifiers and (b) List of set of features associated with each blob, including the location, size and color of the blob.
2) Find the blob identifier corresponding to the maximum size, say blob-i.
3) Find the blob identifier corresponding to the highest grey level, say blob-j.
4) If only the size feature is used, terminate the process, otherwise continue.
5) If only the color feature is used terminate the process, otherwise continue.
6) In this case, the combined size and feature are used for further identification.

```
IF blob-i =  blob-j THEN the Oocyte is blob-i or blob –j
        ELSE
        Trigger a classification program to identify the Oocyte
END IF
```

It is noted that the classification program mentioned above is not useful to this thesis, because the Oocyte is always the largest blob.

(iii) Post process

The post process retrieves the information about the Oocyte, including

1) The co-ordinate of the Oocyte.

2) The Volume (3D) or the Oval (2D) of the Oocyte.

For the program demonstration purpose, the post process may also mark or indicate the position of the Oocyte.

3.2.2 The Pattern Matching Approach

The pattern matching is the procedure for finding which sub-region in the entire image picture best matches the pre-defined pattern. When using the pattern matching approach, the pattern for the oocyte is a rectangular box, which is further defined by [2m] by [2n] pixels (see Figure 3.3). In the present study, m = n = 64. The entire image is defined by [s] by [t] pixels, where in the present study s = 768, t = 576. The matching is measured by a similarity measurement at the pixel level.

In the present study, the similarity measure or correlation measure at the pixel (u, v) is defined as follows:

$$R_{uv} = \frac{N \sum_{i=u-m+1}^{u+m} \sum_{j=v-n+1}^{v+n} M_{(i+m-u)(j+n-v)} S_{ij} - \left(\sum_{i=u-m}^{u+m} \sum_{j=v-n}^{v+n} S_{ij} \right) \sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}}{\sqrt{\left[N \sum_{i=u-m}^{u+m} \sum_{j=v-n}^{v+n} S_{ij}^2 - \left(\sum_{i=u-m}^{u+m} \sum_{j=v-n}^{v+n} S_{ij} \right)^2 \right] \left[N \sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij}^2 - \left(\sum_{i=1}^{m} \sum_{j=1}^{n} M_{ij} \right)^2 \right]}}$$

Where S is the grabbed image of size [s] x [t] pixels. M is the pattern of size [2m] x [2n]. N is the number of pixels in M.

R_{uv} is the correlation measure at the pixel (u, v). The algorithm starts at the upper left corner and moves towards the right, row by row (down). The algorithm for the pattern matching approach is described as follows:

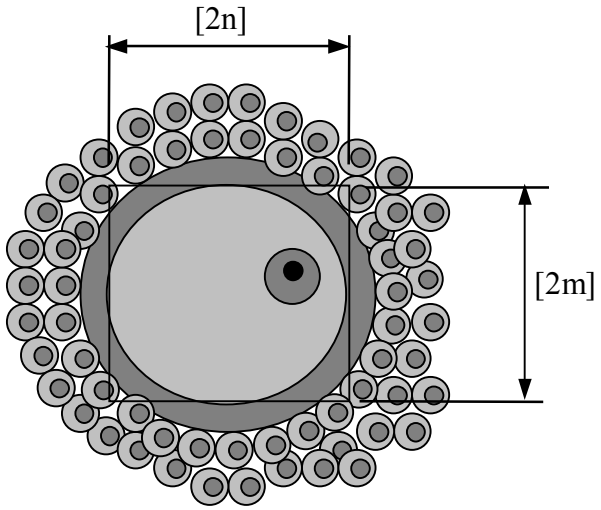


Figure 3.3: Definition of the pattern for the oocyte.

R_{uv} = 0; row-i* = 0; column-j* = 0
For row-i = 1 to s
For column-j = 1 to t do
Calculate R using the correlation measure
IF R > R_{uv}, THEN
    R_{uv} = R
    Row-i* = i
    Column –j = j
END IF

Post condition: The region that best matches the pattern is located (see Figure 3.4).

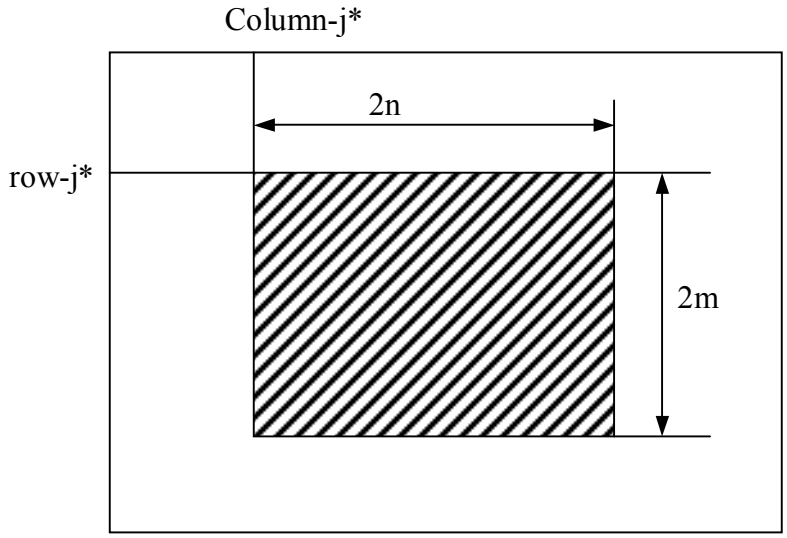


Figure 3.4: The matched region.

3.3 Algorithm of the Holder

The rectangular relationship of the holder is shown in Figure 3.5. The holder has a distinct pattern that is different from other objects. Therefore, it is a good strategy to use the pattern matching approach.

In practice, the Oocyte is first identified. Its area under the whole image is known. For the holder, the pattern matching area is reduced to the total image area minus the Oocyte area.

The pattern box for the holder is defined as shown in Figure 3.5.

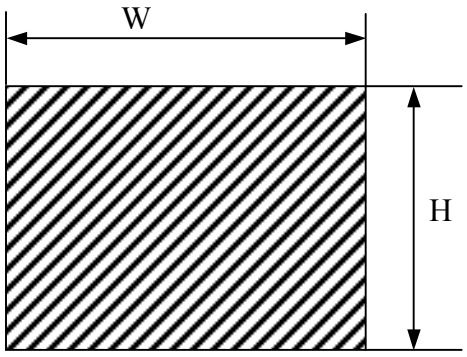


Figure 3.5: The pattern box of the holder.

3.4 Algorithm for Injector

The micro injector is different from the cell and holder because it appears as a long and narrow region in the image. Therefore, it is not suitable to use the pattern matching method to recognize the micro injector because of the low efficiency and poor accuracy. As a result a method based on the binary morphological operation is used to recognize the injector (Xudong et al., 2001).

The image S captured by the camera was binarized. This resulted in quite a few tiny blobs in the binarized image B. These blobs were the products of noise in the image and harmful to the successful processing of the image. They were therefore removed using the open operation. Figure 3.6 helps to explain the algorithm implemented.

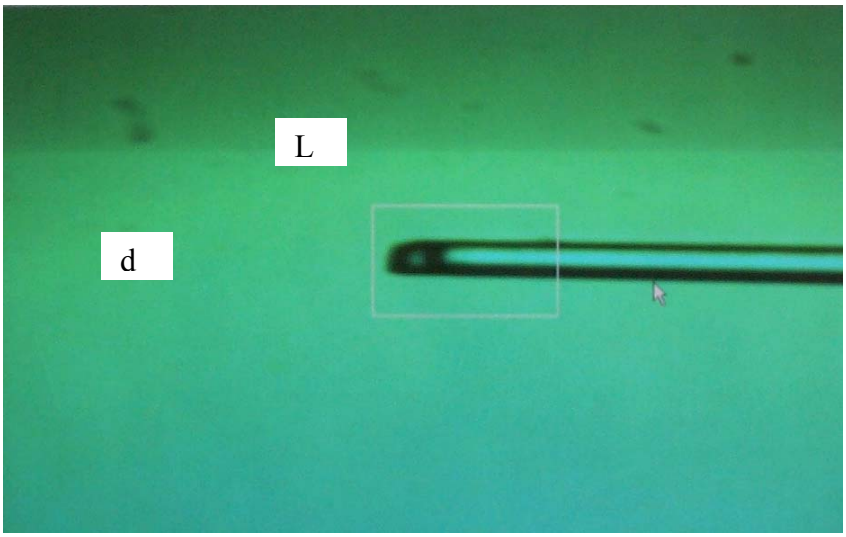


Figure 3.6: Injector image with the dimensions.

The injector has the shape of a cylinder. A threshold of E_0 is obtained from the actual dimensions of the microinjector. In this application, the threshold s defined

$$E_0 = \frac{L}{d}$$

where L is the length, d is the diameter, and E_0 is a coefficient. In this setting d is fixed, yet can vary. Through some initial tests, E_0 was set to 10, therefore L = 10d.

For every blob in B, if its E value is greater than E_0 it is regarded as corresponding to the microinjector, otherwise it is erased. Thus, at the end of the process, only one blob is left in the binary image if E_0 is carefully selected. Finally, a scan line is used to obtain the coordinates of the left point of the blob. This point corresponds to the tip of the micro injector.

3.5 Implementation

The complete algorithm was implemented using the C++ programming language and an image processing software system called Matrox® Imaging Library (MIL) version 6.1. The Matrox® provides a library of functions to manipulate the image at various levels (the lowest level is a single pixel). Matrox® functions can be called by C++ code. The main program was the C++ code with the Matrox® functions embedded into it.

3.5.1 Identification of the Oocyte

(i) Preprocessing of the image

Preprocessing is the name used for operations on images at the lowest level of abstraction. Both input and output are intensity images.

a. Removal of the noise on the image file

Real images are often degraded by some random errors (noise). Noise can occur during image capture, transmission, or processing, and may be dependent on, or independent of, image content. Such noise cannot be removed very easily at the source. Therefore, pre-processing is required to improve the image as much as possible without affecting the useful information. Image smoothing discussed in section 2.3.8 was applied.

b. Binarize the image

A binarizing operation reduces an image to two greyscale values: 0 and the maximum value in the image (for example, 255 if the image is 8-bit). Binary images are useful when trying to identify geometrical patterns and objects since they are not cluttered with shading information. The binarizing operation is performed by comparing each pixel value in the image against the threshold value.

Thresholding is the transformation of an input image *f* to an output binary image *g* as follows:

$$g(i,j) = \begin{cases} 1, & for \quad f(i,j) \geq T \\ 0, & for \quad f(i,j) < T \end{cases}$$

Thresholding images means reducing each pixel to a certain range of values. Some operations can be performed more efficiently on thresholded images. Thresholding is computationally inexpensive and fast. It is widely used in simple applications.

The correct threshold selection is crucial for a successful binarized image. This selection can be determined interactively, or it can be the result of a threshold detection method. In the cell image process, two methods are employed. One is determined by the user interactively and the other is based on histogram shape analysis.

If an image consists of objects of approximately the same grey-level that differ from the grey-level of the background, the resulting histogram is bi-modal. Pixels of objects form one of the peaks, while pixels of the background form the second peak. In this case, the threshold value is set to the minimum value between these peaks.

c. Removal of small particles by the morphological operations

Two morphological operations are used: opening and closing. Erosion followed by dilation creates an important morphological operation called opening. The opening is mathematically defined as:

$$J^o = OPEN(J, B) = DILATE[ERODE(J, B), B]$$

The closing is defined mathematically as:

$$J^c = CLOSE(J, B) = ERODE[DILATE(J, B), B]$$

Opening and closing with a standard structuring element was used to eliminate specific image details smaller than the structuring element. Closing connects objects that are close to each other, fills small holes, and smoothes the object outline by filling narrow gulfs. On the other hand, opening removes small particles.

d. Removal of the holder and the injector

In the image file, the holder and injector are different in shape from the cells. The holder and injector are like rectangles while the cells are like circles. The holder and injector have very large elongation values, while the cell elongations are about 1.

According to this feature, it is easy to remove the holder and injector from the image. In the implementation, those blobs whose elongation was larger than a specified value, say 2.0, were removed, resulting in the image that contained cell blobs. Then the blobs that remain are cells.

(ii) Main processing

The simplest and most natural feature of a blob is its area, given by the number of pixels of which the blob consists. The perimeter is then the total number of pixels sides along the blob edges including the edges of the holes.

Once a binary image of the cell of interest is obtained, the center and boundary can easily be determined. The center is simply computed as the center of mass of the foreground component:

$$CogX = \sum_j x_{f,j} / n_f$$
$$CogY = \sum_j y_{f,j} / n_f$$

where $x_{f,j}$ and $y_{f,j}$ are the coordinates of the *jth* pixel of the foreground component and n_f is the size of the foreground component.

The following Matrox functions were used to realize these goals. The MblobReconstrcut() function was used to erase the blobs attached to the border of the image window and to fill the small holes in the blobs.

In this code, the ProMode parameter M_4_CONNECTED was chosen to separate the blobs that are attached to each other. The MblobSelectFeature() function was used to select the interesting features such as the area, perimeter and coordinate value.

The MblobSelect() function was used to delete all blobs that were not of interest such as those with a small or a very large area, and those that were not circular. Here the operation mode M_DELETE was selected to delete blobs that were of no interest from the original image.

The MblobGetResult() function was used to obtain the chosen features and save those values for the corresponding arrays. In the pattern matching approach, the following Matrox functions were used.

The MpatAllocModel() function was used to allocate a pattern matching model from a source image. In such a case, the mouse was used to select the desired cell as a pattern model. The width and height of the cell model selected were 128 pixels; that is, m = n = 128.

The specified models required search parameters for the positional accuracy, search speed, certainty level, and acceptance level had to be set before the pattern search process. These parameters were set using the MpatSetAccuracy(), MpatSetSpeed(), MpatSetAcceptance(), and MpatSetCertainty() functions.

In the implementation, the certainty level was set to 80% because a very good result was obtained using this value after a trial and error test. If the correlation between the image and the model was equal to or greater than the certainty level, a match was assumed without looking elsewhere in the image for a better match.

To increase the speed of the search process, the search region of a model was restricted to a specified area in the target image using the MpatSetPosition() function.

After that, the MpatFindModel() function was called to search the model in the specified region and the matched model's position was obtained using the MpatGetResult() function.

(iii) Post processing

After the main processing, all of the interesting information was obtained and stored in the arrays. The purpose of the post processing was to highlight the object identified and to show its coordinates.

These features were displayed in a list box in the main window using the C++ code. The marks of the coordinates of the cells were drawn using the Matrox software MgraLine() function. The selected cells were shown in the image field using the binary image format. This also indicates that the position or location of the cells had been identified using the C++ code.

3.5.2 Example

Figure 3.7 shows a graphical interface of the program. There are several menu items shown in the right area. The menu int Digitizer could be triggered to load the cell cluster image, which includes the injector and the holder. The feature button triggered the algorithm to be executed on the displayed image. The location, area and size of the displayed cell could also be determined by clicking the location, area, and size button, respectively.

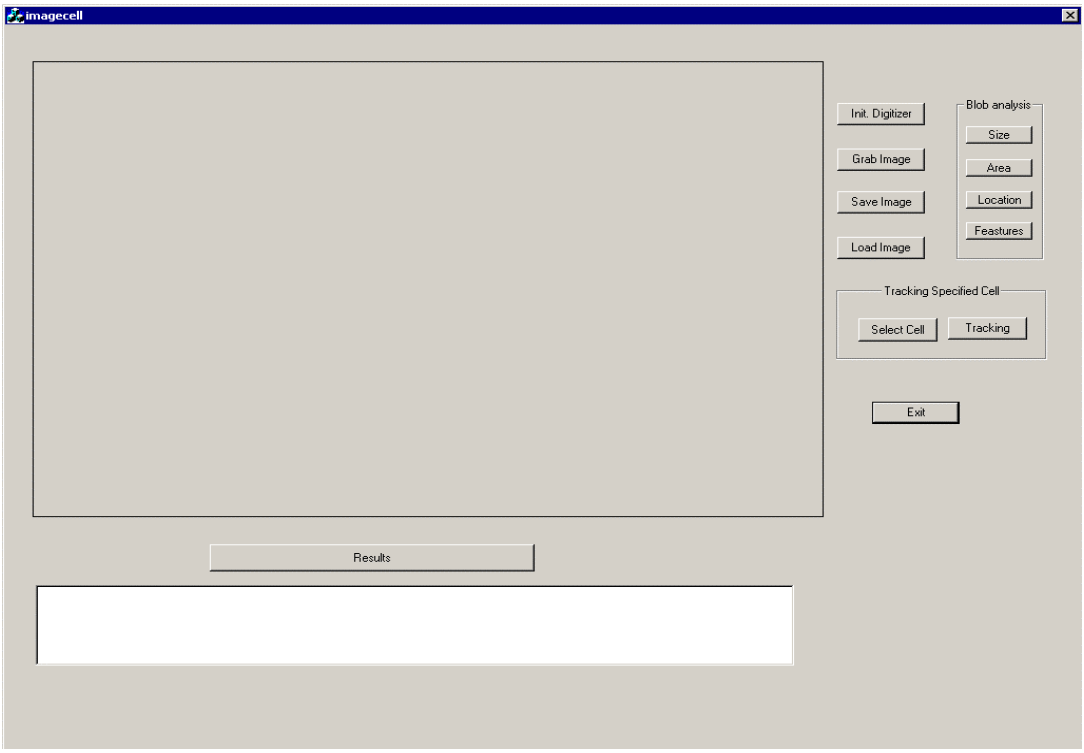


Figure 3.7: The graphic user interface.

Figure 3.8 shows the image of a single cell as it appeared on the graphic user interface

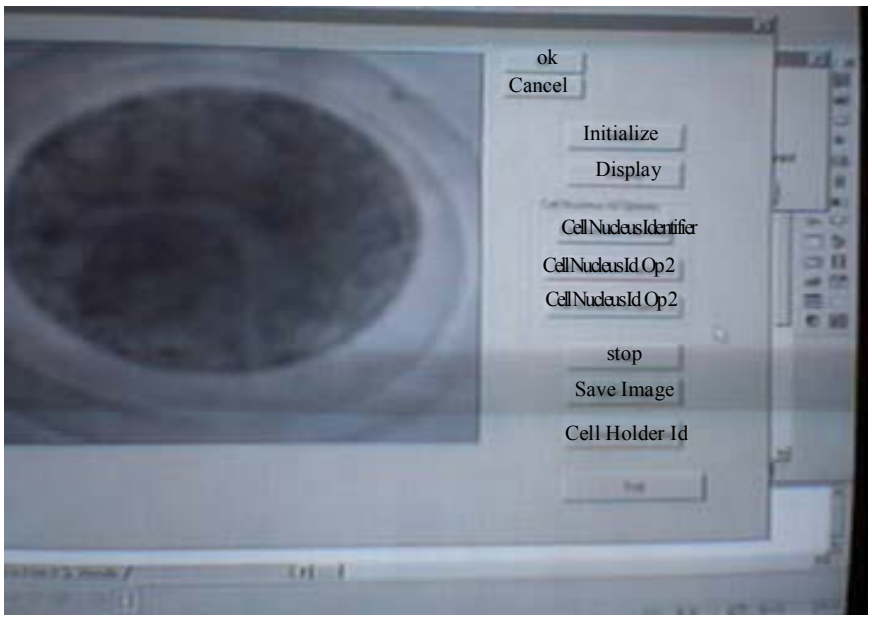


Figure 3.8: Image of a cell grabbed by the application program.

3.5.3 Implementation of the Algorithm for the Injector

As mentioned in Section 3.4, the injector has a special shape; that is, it has a pointed or sharp end. As a result, the pattern matching method cannot be used to recognize it. In this study, the feature matching approach is used.

Based on the special shape, the feature chosen was the M_ELONGATION:

M_ELONGATION=length/breadth

where the length is a measure of the true length of the injector, and the breadth is a measure of the true breadth.

In the implementation, the captured image was first binarized using the MimBinarize() function. Then the MimOpen and MinClose() functions were used to remove the noise and the small holes. After that, the MbolbSelectFeature() function was used to pick up the selected feature. In the selected blobs, only those whose elongations were greater than 10 (E_0) were calculated and recorded. As there was only one injector in the image, one blob was obtained. Once the injector blob was obtained, the tip position of the injector was obtained by using the MblobSelecFeature() and the MblobGetResult() functions with the parameters M_BOX_X_MIN and M_Y_MAX_AT_X_MIN.

Figure 3.9: The injector.

3.5.4 Implementation of the Algorithm for the Holder

The pattern matching approach was used to identify the holder using the procedure described in Section 3.5.1.

As the holder was the same size as the cell, the model size for the holder was selected as 128*128 pixels; that is, W = H = 128. Since, the holder had a very clear image in the microscopy, in the implementation the certainty level was set at a higher level of 90%.

In order to increase the speed of the search process, the search region of a model was restricted to a specified area in the target image. The MpatSetPosition() function was used for this purpose. After that, the MpatFindModel() function was called to search the model in the specified region by evaluating the correlation function at every pixel in the search region. This procedure was continued until the correlation was more than 90%, and the location of the matched holder was returned.

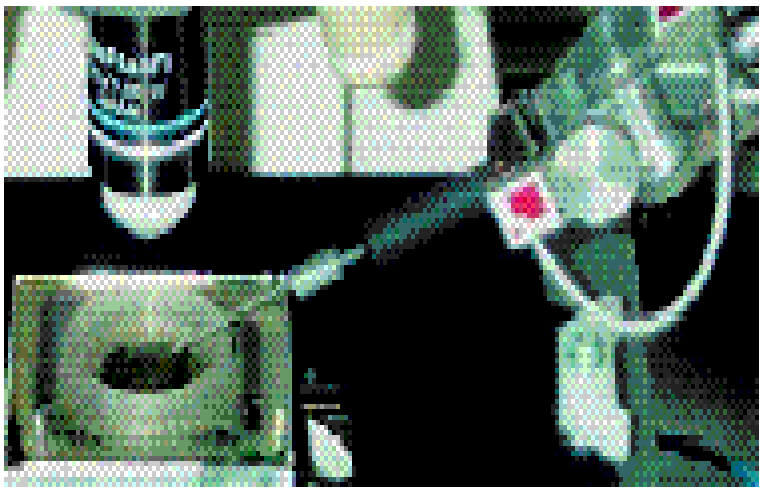


Figure 3.10: The holder.

3.5 Summary

The algorithms, as well as their implementation for the oocyte, the holder and the injector were described in this chapter. Both the feature matching method and the pattern matching method were applied to the algorithm of the oocyte. In the case of the holder, the pattern matching method was applied. For the injector algorithm, a special method that consists of the morphological operations (the opening and closing) was used because the injector had a thin pointed end.

Chapter 4

Experimental Setup, Results and Discussions

4.1 Introduction

In this chapter, the experiment to test the algorithms developed in the previous chapter will be described. The purpose of this test was to evaluate the performance of these algorithms. An overview of the experiment set-up is given in Section 4.2. The method used to measure the performance of the proposed algorithms is given in section 4.3. Sections 4.4, 4.5 and 4.6 respectively, contain description of the results obtained for the cell algorithm, holder algorithm and injector algorithm. The conclusions are given in Section 4.7.

4.2 Experimental Setup

4.2.1 Apparatus

The apparatus used in the experiment included:

- Optical microscope (model: 1X70-S1F2; manufacturer: Olympus Optical Co. Ltd);
- CCD camera (model: TK 1270; manufacturer: JVC);
- Image grabbing software (Model: Matrox® Imaging Library version 6.1; manufacturer: Matrox Electronic System Ltd);
- Processing board and image processing computer (model: Genesis; manufacturer: Matrox Electronic System Ltd).

The CCD camera, which was used to capture the image from the microscope, was attached to the base of the microscope. The captured image was sent to the image processing computer through the image processing board. A Visual C++ ® GUI with MIL embedded software was used to process these images. The arrangement of the apparatus is shown in Figure 4.1.

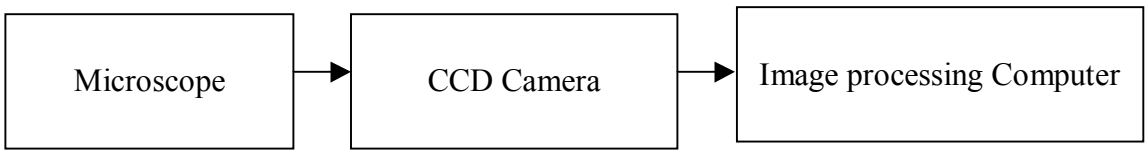


Figure 4.1: Block diagram of the image processing module.

The computer used for the development of the algorithm had the following configuration:

- Intel Pentium 4 CPU 1700MHz with 512M RAM;
- Windows 2000 operating system;
- Matrox® Imaging Library;
- Visual C++ ®6.0.

All the programs were realized using Visual C++ ® to supply a user-friendly interface and the Matrox® Imaging Library used to fulfill the image processing functions. In particular, the Matrox code was embedded in the C++ code.

4.2.2 Materials

The cell cluster, which came from the tissue of a cow, was provided by the Department of Veterinary Biomedical Sciences of the Western College of Veterinary Medicine College, University of Saskatchewan. The cell images are shown in Figure 4.2. The dark portion of the cells indicates the oocytes where the nucleus resides. This is the region that the application program attempts to identify.

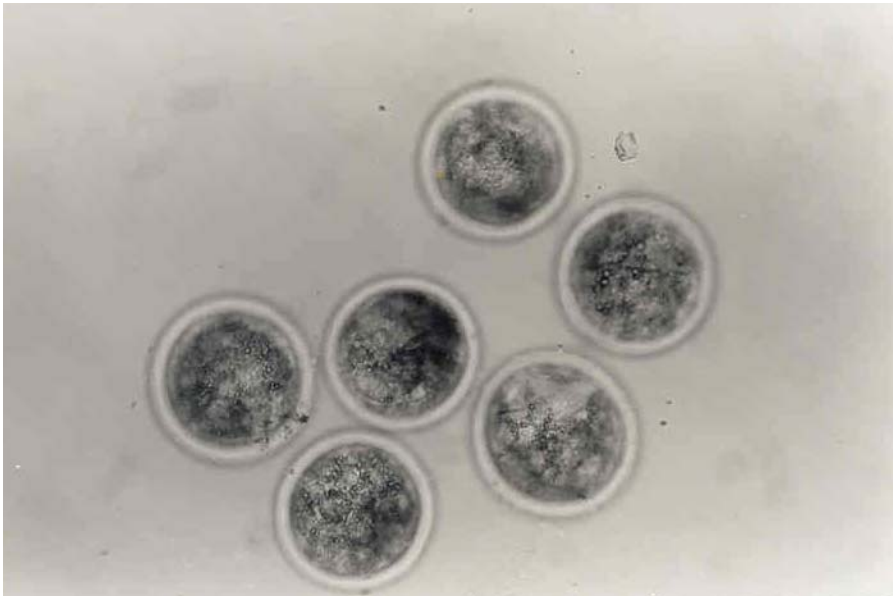


Figure 4.2: Digital image of multiple cells.

The oocyte region of the cell can be clearly seen in the digital image shown in Figure 4.3, which distinctly shows the oocytes. This is the region where the genes are injected by the cell auto manipulator.

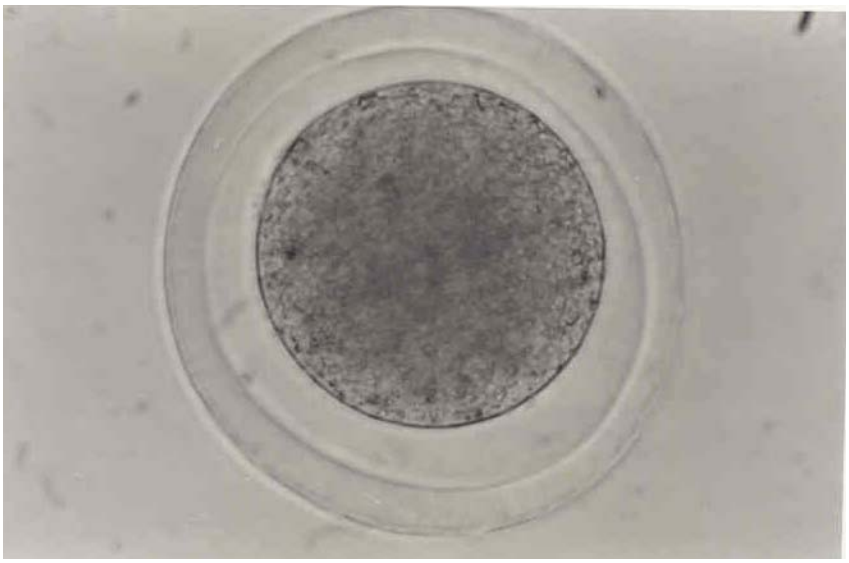


Figure 4.3: Digital image of a single cell.

The images of the holder and the injector captured by the camera are shown in Figure 4.4 and Figure 4.5, respectively. The holder is used to hold the cell in place for manipulation and injection of the genes, while the injector is used for manipulating and the injecting genes into the cell.

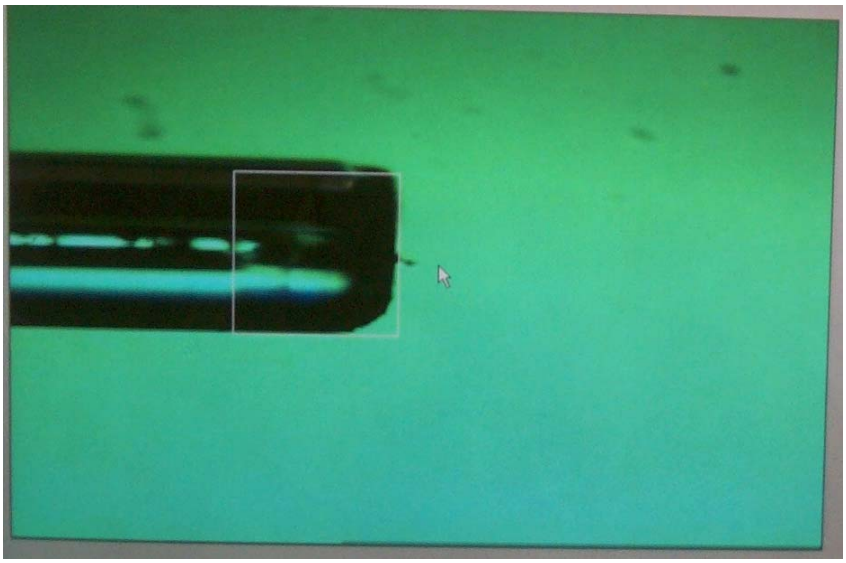


Figure 4.4: Digital image of a holder.

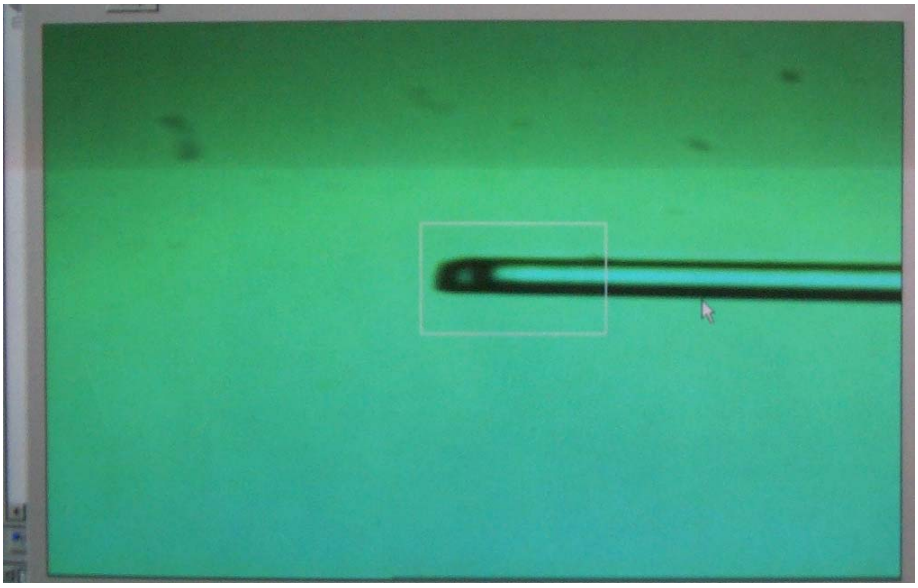


Figure 4.5: Digital image of a needle.

4.3 The Method

To measure the performance of the proposed algorithms the time needed for the algorithms to complete the task was recorded. The following program code was developed for this purpose.

```
CTime startTime. endTime;
int Begining_of_Iteration=1;
int End_of_Iteration=100;
int Steps = 1;
ofstream myFile("c:/Blessing/Results/ProcessTime20.xls");
=====
Initialization of the system
```

```
=====
startTime = CTime::GetCurrentTime();
for (int k = Begining_of_Iteration;k <= End_of_Iteration;k=k+Steps)
{
Body of the main program of the feature extraction process
}
endTime = CTime::GetCurrentTime();
m_StartTime=
((int)startTime.GetHour()*3600)+((int)startTime.GetMinute()*60)+((int)startTime.GetSe
cond());
 m_EndTime
=((int)endTime.GetHour()*3600)+((int)endTime.GetMinute()*60)+((int)endTime.GetSec
ond());
m_Diff = (float)(m_EndTime-m_StartTime);
m_AveTime = (float)m_Diff/(float)(k-1);
UpdateData(FALSE);
myFile << "Iteration"<< "    "<< k-1 << "       " << "Average Time"<<"    "<< m_AveTime
<< endl;
```

4.4 Cells Algorithm Results

Due to the fact that the feature color and size were used to identify the nucleus, two buttons were developed, representing color and size, respectively. Their combination was made by a third button.

When these buttons were pressed, the algorithms developed in Chapter 3 were triggered and the program started. The output of the button representing the size feature of the cell nucleus or oocyte is shown in the table below:

Table 4.1 Result of the Size Feature Algorithm

No	Area	Perimeter	Diameter	Position
1	9713	414	139	494, 63
2	14391	452	133	272, 116
3	14656	468	154	627, 182
4	11432	430	137	432, 202
5	9930	399	125	179, 260
6	10047	405	131	319, 364

It should be noted that the results listed in Table 4.1 were based on the size feature. When the oocyte was located, the following were found using the feature matching algorithm: the area, perimeter, diameter and location of the cell.. If one needs to find the physical size of the cell, the calibration of the camera was needed before the implementation of the processing. When the color button was clicked, the output generated a histogram of the threshold intensity.

A histogram is the intensity distribution of pixel values in an image and is generated by counting the number of times each pixel intensity occurs. This information is useful to select a threshold level when binarizing an image.

The results obtained from the histogram were used to obtain an optimum threshold value for the algorithm which was used as one of the parameters in the MimBinarize() function.

Several threshold values were tested and different histograms were obtained for the respective threshold values. The experiment was carried out on digital images containing a single cell as shown in Figure 4.3.

For a threshold value of 20, the histogram shown in Figure 4.7 was obtained

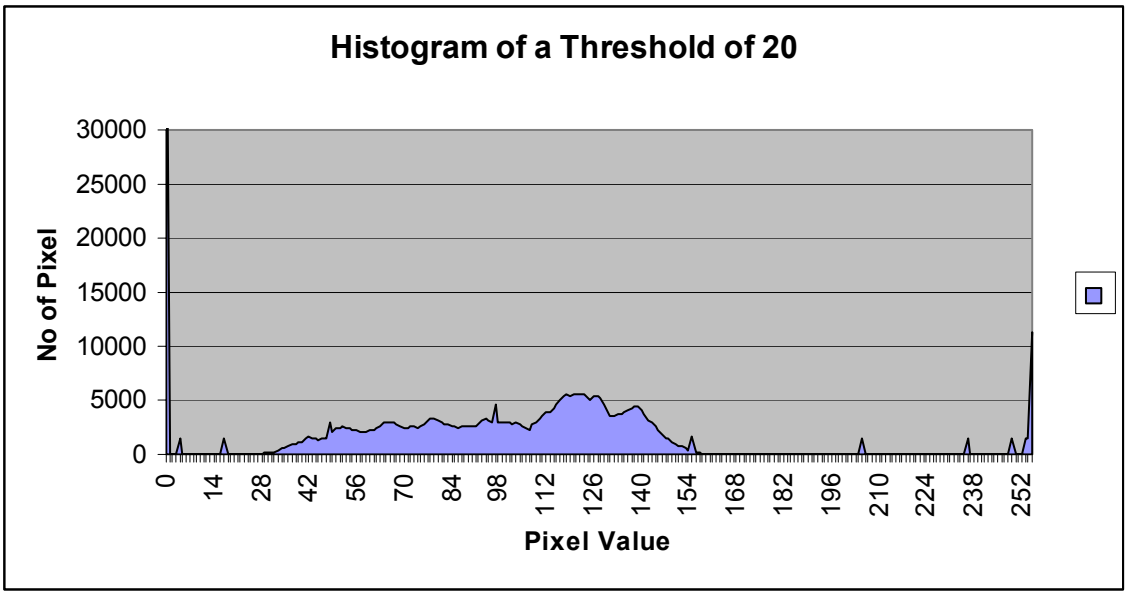


Figure 4.7: Histogram of threshold intensity value of 20.

When the threshold value was increased to 120, the histogram shown in Figure 4.8 was obtained.

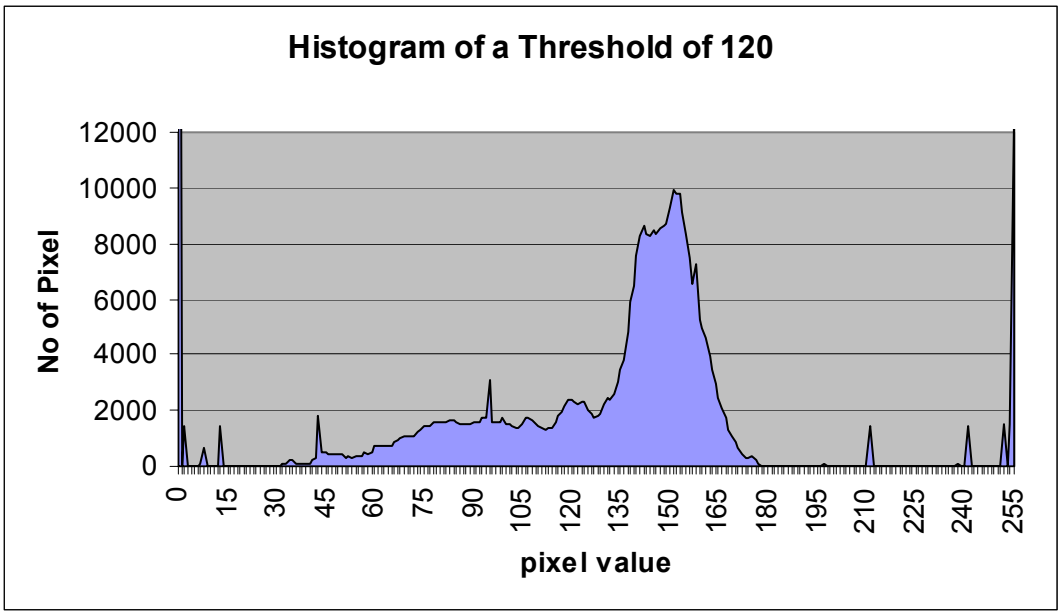


Figure 4.8: Histogram of threshold intensity value of 120.

For a threshold of 140, the histogram shown in Figure 4.9 was obtained.

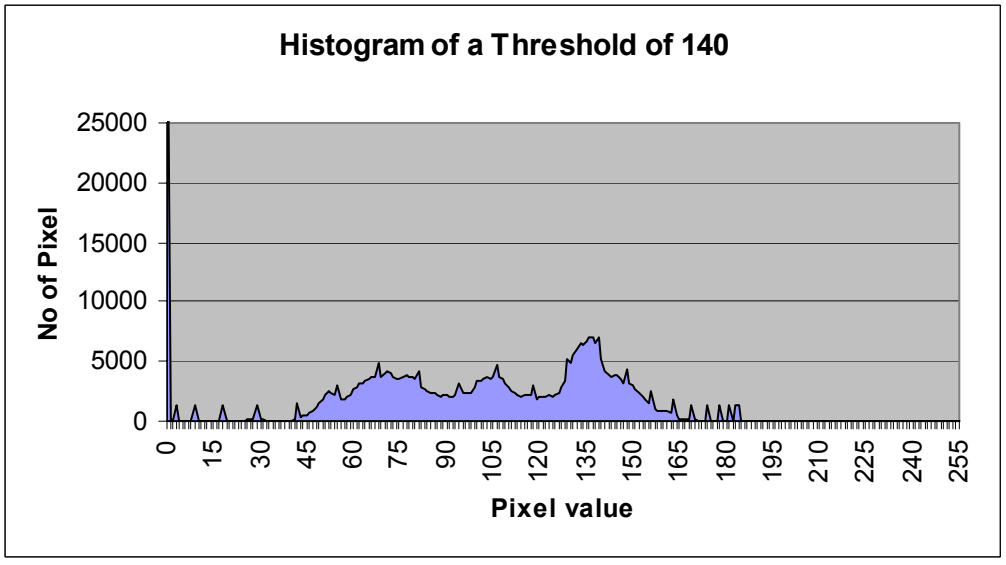


Figure 4.9: Histogram of threshold intensity value of 140.

Finally a threshold value was increased to 200 and the histogram in Figure 4.10 was obtained

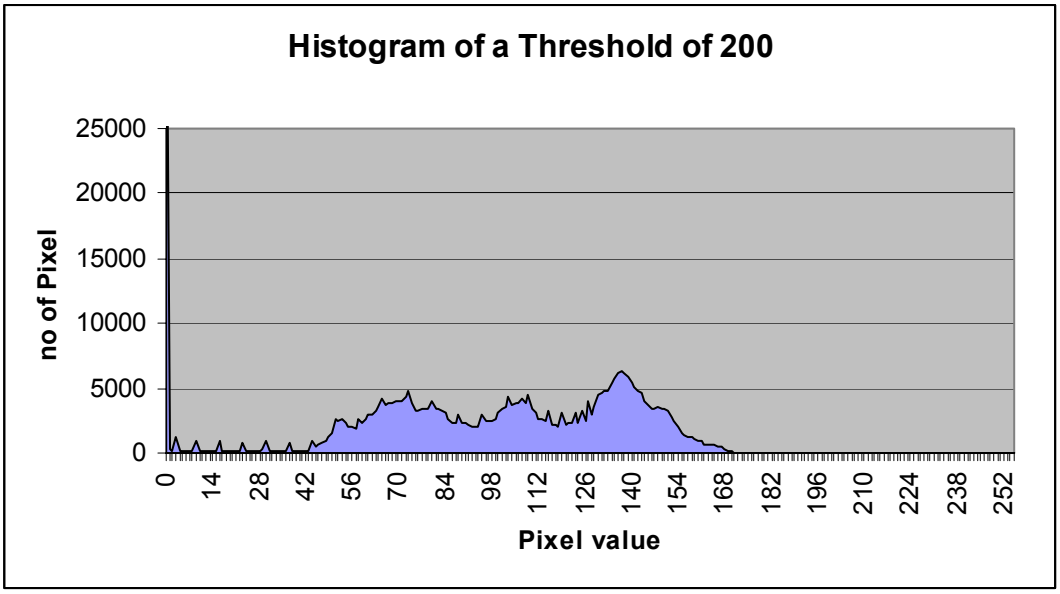


Figure 4.10: Histogram of threshold intensity value of 200.

From the three values tested, it was observed the suitable threshold could be set between 90 and 95.

When a threshold value of 140 was implemented on single and multiple cells, the results shown in Figures 4.11 and 4.12 were obtained for both single cell and multiple cells, respectively. The oocyte is indicated in Figures 4.11 and Figure 4.12 by the white circular color surrounded by a black background.

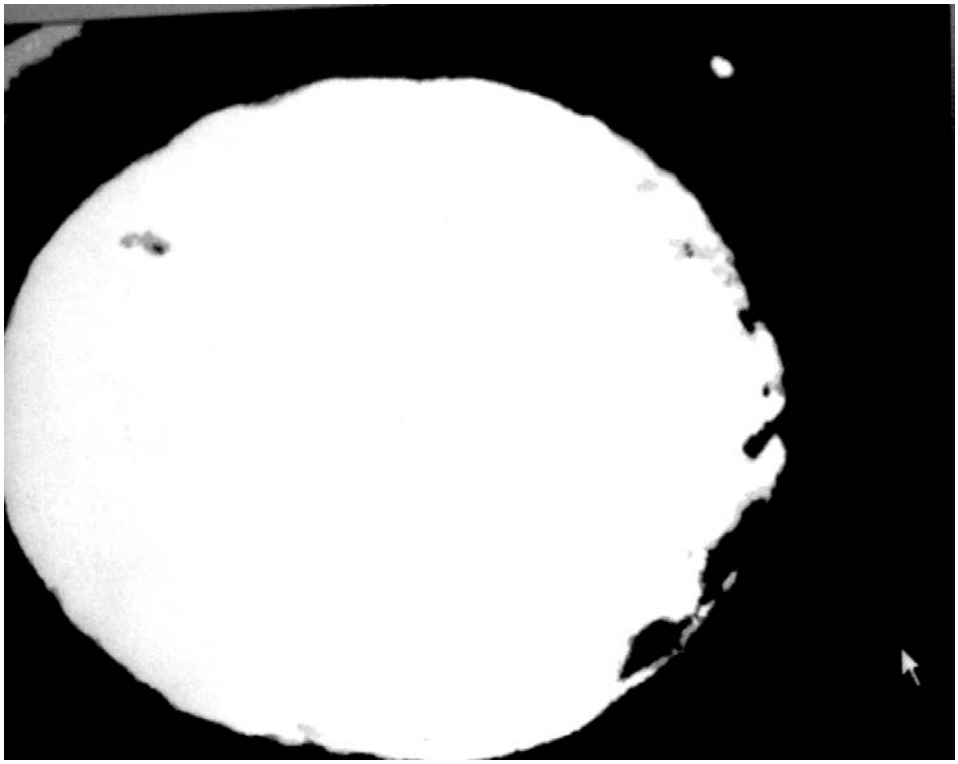


Figure 4.11: The result of using the grey level intensity on single cells.

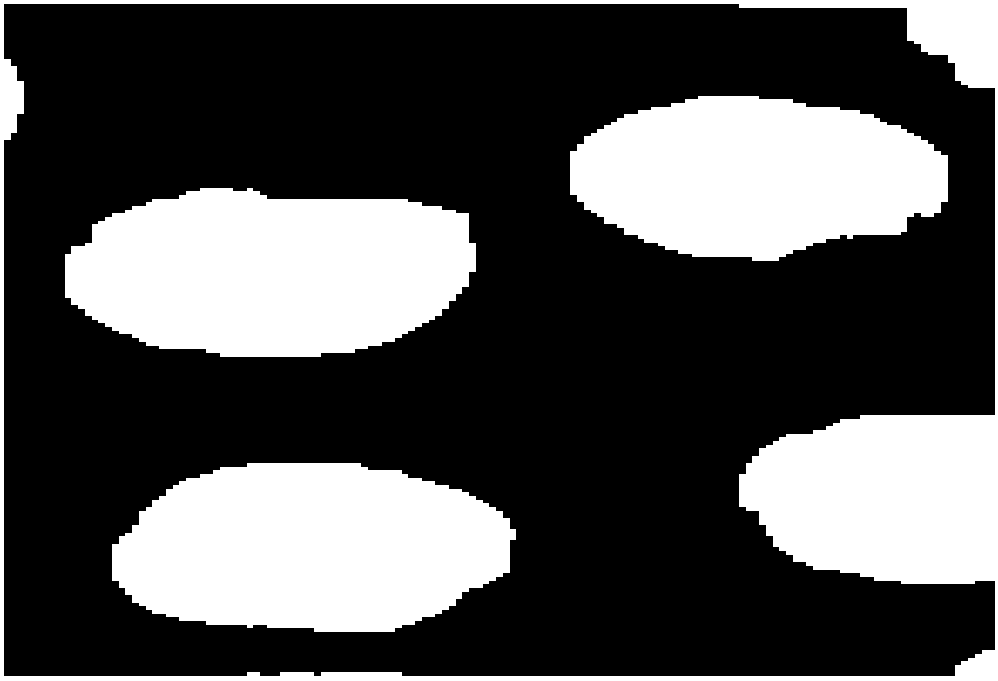


Figure 4.12: The result of using the grey level intensity on multiple cells.

The pattern matching technique entails comparing the image grabbed by the camera and the image of the nucleus of the cell stored in a buffer. Then, both images are correlated in order to find a match and a rectangle is drawn at the position where a match is found. Figure 4.13 shows the result of using the proposed algorithm.

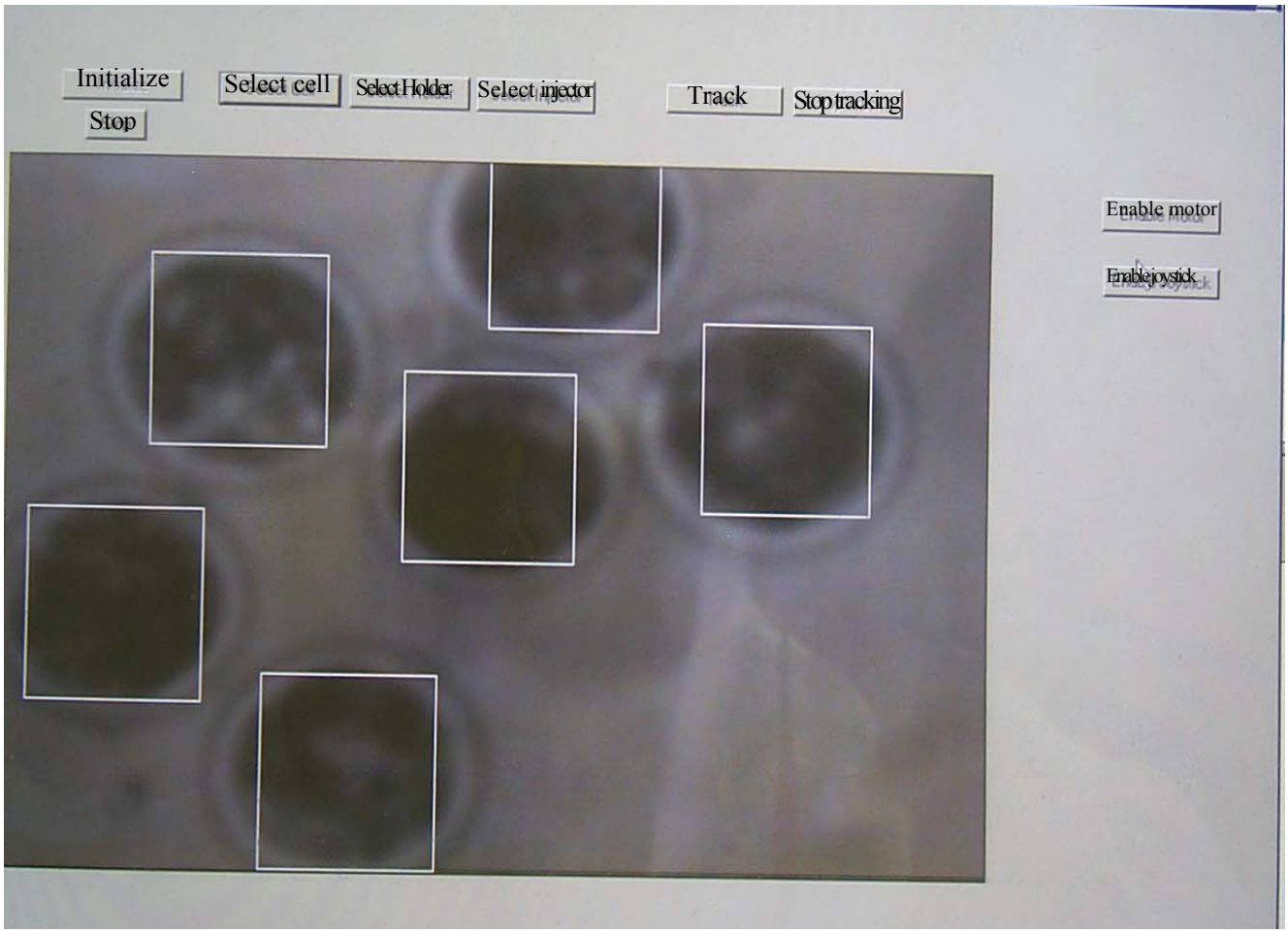


Figure 4.13: The result pattern matching method on cells.

The performance of the algorithm was measured by the time it took to complete the identification process. This was achieved using the following procedure to measure the performance of the algorithm in identifying the nucleus without locating the position of the nucleus.

Since the system clock could not record time in fractions of a second, the program had to be executed for a number of iterations. For this experiment the iteration began at 1, and ended at the 100th iteration with a step of 1. The start time was recorded before the program was executed and the end time was recorded after 100 iterations of the execution of the algorithm. Tables 4.2 and 4.3 show the time for executing 100 and 500 iterations on images containing single cells and six cells respectively.

Table 4.2 Time Duration of the Algorithm for 100 Frames

Parameters	Single Cell (sec)	6 Cells (sec)
Size	0.11	0.12
Area	0.11	0.12
Location	0.12	0.11
Features	0.12	0.12

Table 4.3 Time Duration of the Algorithm for 500 Frames

Parameters	Single Cell (sec)	6 Cells (sec)
Size	0.11	0.12
Area	0.11	0.12
Location	0.10	0.11
Features	0.12	0.11

It should be pointed out that 60% of the time that was spent by the algorithm was spent on grabbing the image and about 40% of the time was spent on finding the features.

Increasing the speed of the grab procedure, or using parallel processing for the grab, will improve the algorithm. For the pattern matching method, the time duration for processing 100 frames of images for different model sizes is listed in Table 4.4. It can be seen from the results that, as the size of the cell model decreased, the time decreased and vice versa. In order words, there is a direct relationship between the model and the time used in executing the algorithm.

Table 4.4: Time Duration using the Pattern Matching Algorithm on Cells

Model Size	Time (sec)
148 x 148	6.71
138 x 138	6.93
128 x 128	6.70
118 x118	6.76
108 x 108	6.70
98 x 98	6.70
88 x 88	6.67
78 x 78	6.69
68 x 68	6.67

To determine the sensitivity of the pattern size, the above experiment for the pattern matching method on the Oocyte was conducted but no sensitivity was observed.

However, when the size of the model was greater than 148 by 148, the algorithm would not recognize the Oocyte properly. As well, when the size was less than 68 by 68, the model was too small to recognize the entire region of the Oocyte.

4.4.1 The Results of Identifying and Finding the Location of the Nucleus

It has already been explained that to locate the position of the nucleus of the cell, the nucleus is treated as a blob and two functions are called. These functions are:

a) MblobSelectFeature(FeatureList,M_AREA);
b) MblobSelectFeature(FeatureList,M_LOCATION);

The function MblobSelectFeature() has a list of feature parameters, two of which are M_AREA that enables the area feature to select the blobs of interest and M_LOCATION that locates the position of the nucleus of the cell. However the result is stored in a buffer. The function that stores the result is MblobAllocResult(MilSystem,&BlobResult). From this result the coordinates of the position of the cell is obtained by the function:

a) MblobGetResult(BlobResult,M__LOCATION_X+M_TYPE_LONG,CogX);
b) MblobGetResult(BlobResult,M__LOCATION_Y+M_TYPE_LONG,CogY);

However, the Cartesian positions of the nucleus, with reference to the origin, have to be identified. This was achieved by calling a function that would carry out the task of drawing a cross at the location of the nucleus to indicate that the nucleus had been located and identified.

The result of executing the program is shown in Figure 4.12 below.

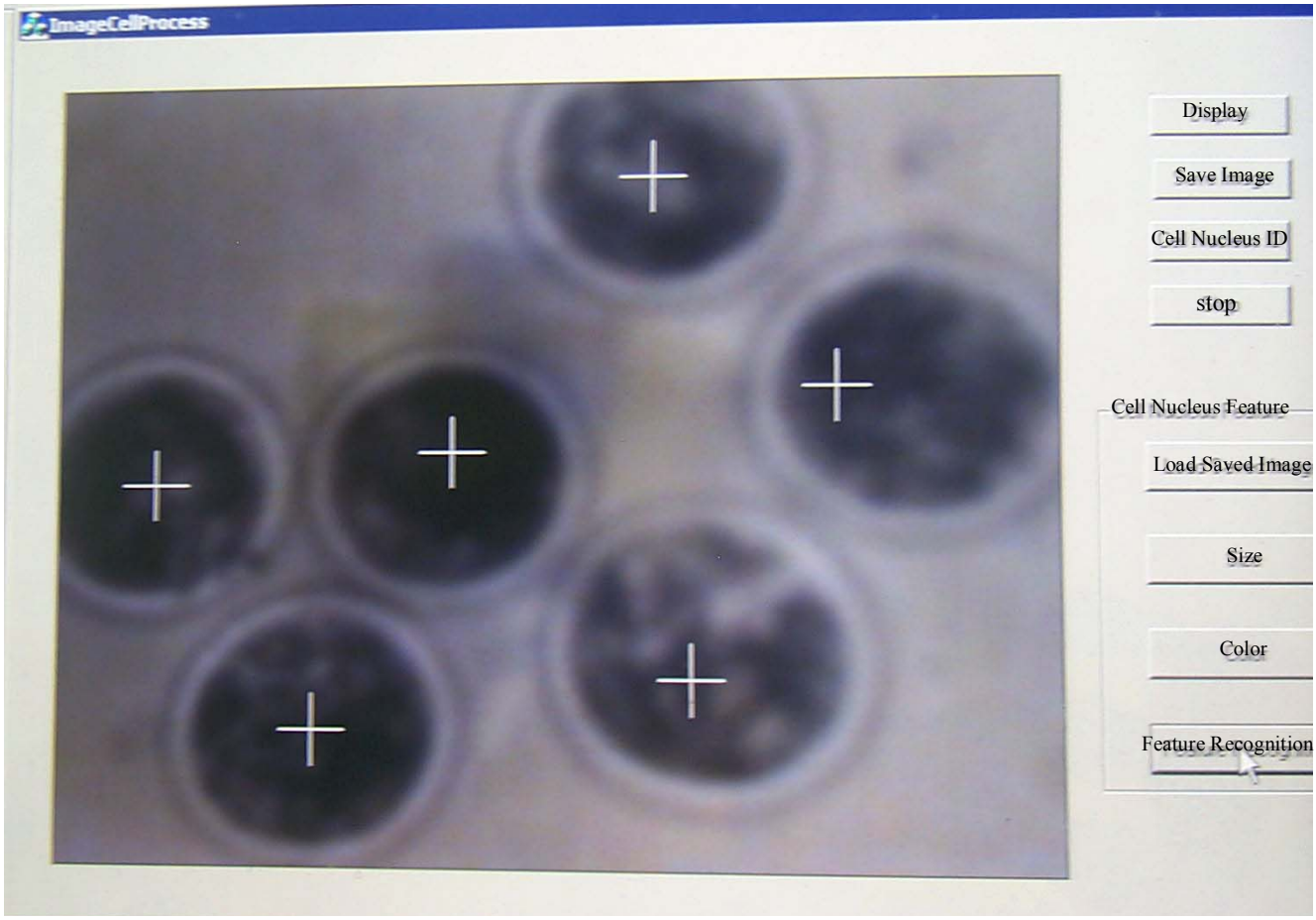


Figure 4.14: Result of locating the position and identifying multiple cell nucleus.

The results shown in Figure 4.14, confirm that the positions for multiple cells have been successfully identified. When the same algorithm was implemented on digital images containing single cells, the results in Figure 4.15 were obtained. This indicates that the algorithm had successfully identified the Oocytes.

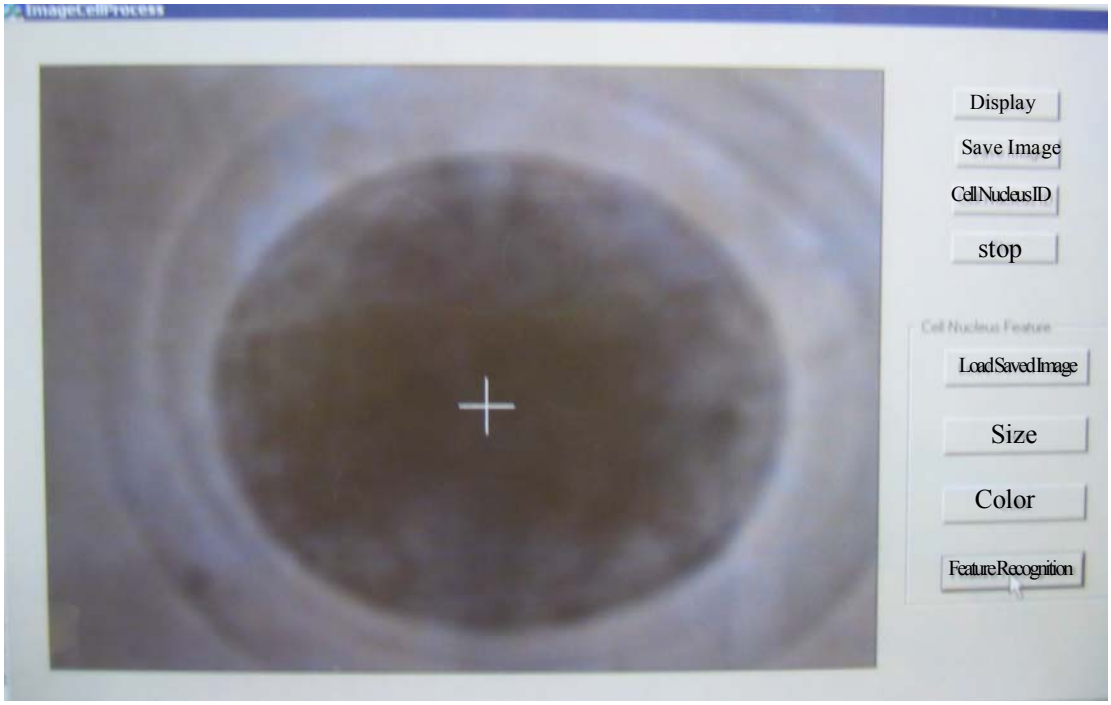


Figure 4.15: Result of locating the position and identification of single cell nucleus.

When a threshold value of 120 was applied to Figure 4.13, the following results were obtained:

0 X Co-ordinates 249 Y co-ordinates 229 Area 1212539456 Diameter is 142109606

When a threshold value was reduced to 20, no recognition was observed; therefore, there was no location of the position. When the threshold was increased to 200, the following results were obtained, indicating that only one position was located:

0 X Co-ordinates 255 Y co-ordinates 239 Area 1215299584 Diameter 1142671494

For multiple cells, when a threshold of 120 was used, only one position was recognized as shown below:

0 X Co-ordinates 227 Y co-ordinates 260 Area 1210869056 Diameter 1142216692

However, an optimum recognition was obtained when the threshold was 90. Table 4.5 shows the results obtained.

Table 4.5: Variation of the Threshold

Positions	x-coordinates	y-coordinates	Area	Diameter
0	424	62	1183252992	1125966882
1	44	264	1179987968	1125401010
2	248	259	1185780224	1126919802
3	423	393	1178461184	1125535167
4	156	423	1183458816	1126080435

This indicates that the threshold plays a large part in locating the position of the cells. Therefore, choosing the optimum threshold is important in recognizing the cells. The histogram gives the range of the optimum threshold.

4.5 Holder Algorithm Result

In Chapter 3 the pattern-matching algorithm was developed for recognizing the holder of the cell manipulation system. This involved comparing the image grabbed by the camera with the image of the holder in a buffer. The pattern matching algorithm was used to find a match in the grabbed image. Once a match was found, a rectangle was drawn at the position of the matched pattern. Figure 4.16 shows the model of the holder used for searching for the pattern on the image.

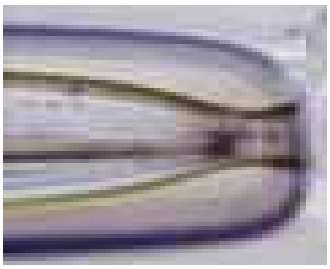


Figure 4.16: The model of the holder.

This algorithm could be implemented because the holder had an almost rectangular end that could match the rectangular box of the stored target image. Figure 4.17 shows the result of the program developed to execute the algorithm.

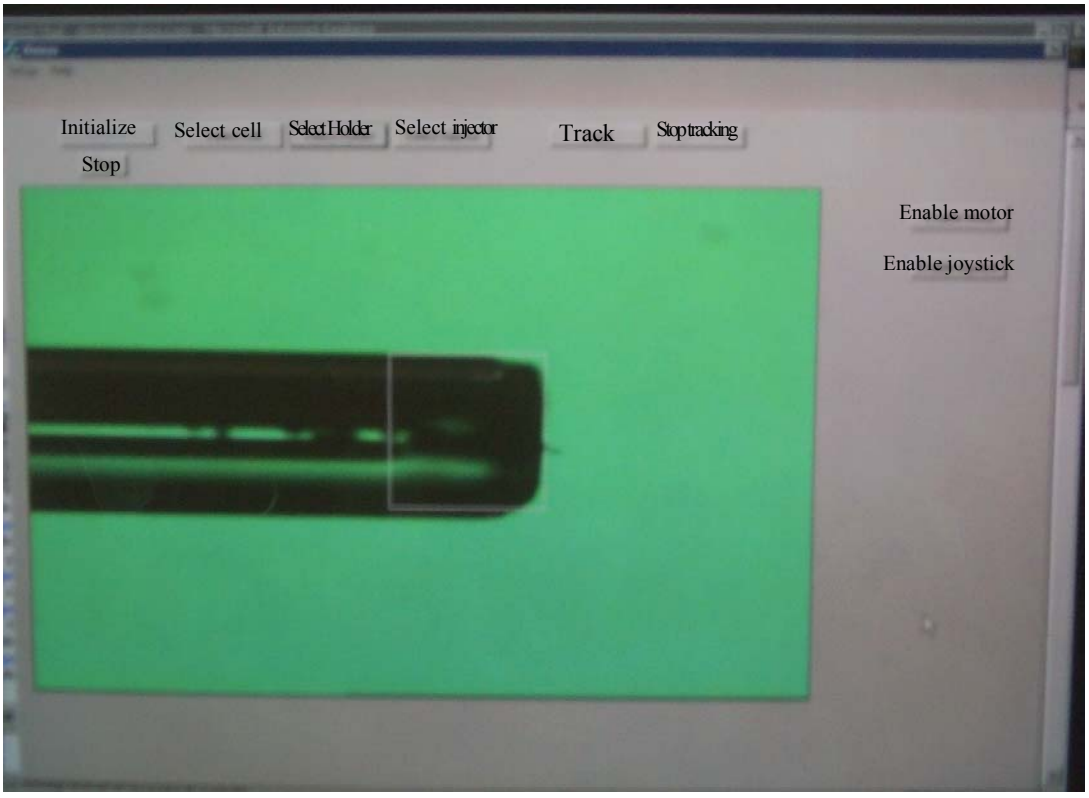


Figure 4.17: Result of pattern matching method on the holder.

The time taken for the program to execute the algorithm was 6.78s/100 frames with an average speed of 14.75 frames/s. For different pattern sizes, the time durations were recorded and listed in Table 4.6. As can be seen from these results listed in Table 4.6, as the size of the cell model decreased, the time decreased and vice versa. In order words there was a direct relations between the size of the model and time.

Table 4.6: Time Duration for the Holder Recognition

Model Size	Time (sec)
148 x 148	6.71
138 x 138	6.93
128 x 128	6.70
118 by 118	6.76
108 by 108	6.70
98 x 98	6.70
88 x 88	6.67
78 x 78	6.69
68 x 68	6.67

4.6 Injector Algorithm Result

The injector required a different approach because of its pointed end. As a result, the pattern matching method could not be used in this case as described in Chapter 3. Rather, a method based on the morphological operation was used to recognize the injector.

The result of implementing the algorithm is shown in Figure 4.18.

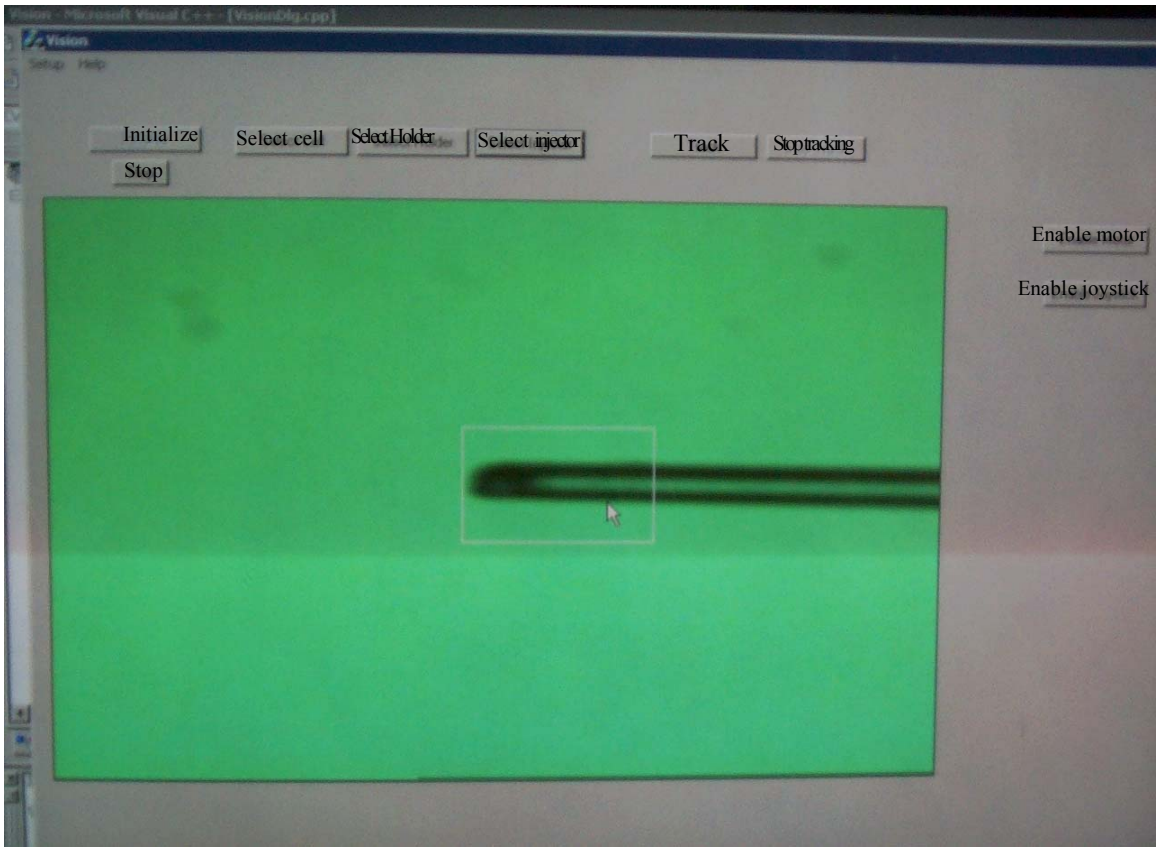


Figure 4.18: Result of the binary morphological operation on the injector.

The time taken for the program to execute the algorithm was 6.72s/100 frames with an average speed of 14.89 frames/s.

4.7 Summary

The feature extraction method and the pattern matching method for cell nucleus identification algorithms have been successfully shown to be able to recognize a cell nucleus. In addition, it has been shown that the feature extraction and the pattern matching methods have the capabilities of recognizing a cell's nucleus and locating the position of the cells. The pattern matching method was used to identify successfully the holder as indicated in Section 4.6. A different approach, called the morphological operation, was used to identify successfully the injector as indicated in Section 4.7. There exists a significant difference in the image identification (Oocyte in this case) between the two methods. The average time for feature matching is 0.11s while that of pattern matching is 6.66s; therefore the feature matching method is 60 times faster than the pattern matching method.

The performance of the algorithm was measured as the time that it took to complete the identification process. The speed of the cell manipulation system was greatly hindered by the image grabber speed. Therefore, increasing the speed of the image grabbing was the problem of concern. In the context of the tracking of cells, the grabbing process needed to be executed once. The effect of this process was very little. In other words, in the case of tracking, the speed was dominated by the image identification process.

The feature and pattern matching methods differed in their algorithms for achieving recognition. However, both approaches could be combined effectively or used separately to achieve a faster image processing of the component of the cell manipulation system. Since the speed of the cell manipulation system was greatly affected by the speed of the image-processing component, a faster recognition algorithm will definitely increase the speed of the cell manipulation system.

Chapter 5

Conclusions and Future work

5.1 Summary of the Thesis

A discussion was presented in this thesis on the issue of tracking biological cells or cell elements for the physical manipulation of the cells. The main concern in this issue was the speed of processing for identification of the objects involved. The main objects that were identified and tracked included:

1) The cell or cell elements
2) The holder and
3) The injector

Although extensive studies have been reported in the literature regarding image processing (Castleman, 1996), the results of these studies could not be applied to the problem directly addressed in this thesis; that is, the fast identification and tracking of the cell holder and injector (Nikolaos et al., 1993). This is due to the special features of these objects. For example, the injector was a thin, long object while the shape of the cell approximated a circle. It is to be noted that the general methodologies for identification of objects are two: (1) the feature matching method and (2) the pattern matching method.

For the identification of the Oocyte in the cell, both the feature matching and the pattern matching methods were applied. In terms of identification speed the results have shown a striking difference in the feature matching and pattern matching methods. The feature matching method was found to be 60 times faster than the pattern matching method.

For the identification of the holder, the pattern matching method was used. In this case of the pattern matching, it was found that the pattern size was sensitive to the processing performance. For the identification of the injector, a method based on the morphological operation was used. The results of the research have shown that all the objects were successfully identified.

5.2 Contribution and Conclusion

The major contributions of this research include:

1) The development of dedicated solutions for the fast identification of the Oocyte, the holder and the injector
2) The development of a method for the identification of the injector, using a novel concept that makes use of the morphological operation.

The major conclusions are:

1) The feature matching method is suited to the identification of the Oocyte in a cell.
2) The pattern matching method is suited to the identification of the holder and the sensitivity analysis has shown that the pattern size selection is important.
3) The morphological operation method is suited for objects with a very thin shape such as the injector.
4) Selecting a suitable threshold value is important in executing the feature matching algorithm.

5.3 Future Work

In the current study, the nucleus of the cell was not identified. The feature of the nucleus is not readily recognized from its surrounding elements, and this creates a challenging problem for the future. Future work should be directed towards the machine vision identification of the nucleus. A possible avenue could be a combination of the empirical and algorithmic methods. The empirical one could be developed using the artificial neural network technique, while the algorithmic method could follow the pattern matching concept.

References:

1. Advance Microsystem Laboratory, University of Minnesota. www.me.umn.edu/division/adv_microsystems/biomanip.htm
2. Allen P.K., Yoshimi B., and Timcenko A., 1991, "Real-time Visual Servoing", *Proc. IEEE Int. Conf. Robotics Automat*. pp 851-856., April.
3. Baldock R. and Graham J., 2000, *Image Processing and Analysis: A Practical Approach,* Oxford University Press, United Kingdom.
4. Butler M., 1991, *Mammalian Cell Biotechnology – A Practical Approach*. Oxford University Press Inc., New York, USA.
5. Carrozza M.C., Dario P., Menciassi A., and Fenu A. 1998, "Manipulation of Biological and Mechanical Micro-Objects Using LIGA-Microfabricated End-Effectors", *Proceedings of the IEEE ICRA*, pp. 1811-1816, May Leuven.
6. Cid-Arregui A. and A. Garcia-Carranca A., 1997, *Microinjection and Transgenesis*, Springer Berlin, Germany.
7. David F. Betsch, 1994, Biotechnology Training Programs, Inc., Iowa State University Office of Biotechnology, U.S.A.
8. Glasbey C.A. and Horgan G.W., 1994, *Image Analysis for the Biological Sciences* Wiley, Chichester, Sussex, United Kingdom.
9. Gonzalez R.C. and Woods R.E., 1992, *Digital Image Processing*, Addison-Wesley, Massachusetts.

10. Graham J., 2000, "Pattern Recognition: Classification of Chromosomes" Oxford University Press Inc., New York, USA.

11. Hogan B., Beddington R., Costantini F. and Lacey E. 1994, *Manipulating the Mouse Embryo: A Laboratory Manual*, Second Edition, Cold Spring Harbor Laboratory Press.

12. Higuchi, Torrii and Yoamamoto Laboratory, The university of Tokyo. http://www.intellect.pe.u-tokyo.ac.jp/research/manipulator/manipulator_e.html

13. Kirk Backstrom, 2000, Technical Report, Advance Engineering Design Lab, Mechanical Department, University of Saskatchewan.

14. Lee S.W. and Wohn K., 1988, "Tracking Object by a Mobile Camera", Dept of Computer and Information Science, University of Pennsylvania, Tech. Rep MS-CIS-88-97, November.

15. Matrox® Image Processing Group, Matrox® Imaging Library User Guide 1997, Matrox® Electronic Systems Ltd., Chapter 12, pp 154-157.

17. Nikolaos P. Papanikolopoulos, Pradeep K. Khosla, 1993, " Visual Tracking of a Moving Target by a Camera Mounted on a Robot: A Combination of Control and Vision" *IEEE Transaction on Robotics and Automation*, Volume 9, No 1, pp. 15-35, Febuary.

18. Papanikolopoulos N.P., 1995, "Selection of Features and Evaluation of Visual Measurements During Robotic Visual Servoing Tasks," Journal of Intelligent & Robotic Systems: Theory and Applications, Vol. 13, pp 279-304.

19. Research Instruments Ltd., Kernick Road, Penryn Conwall, TR10 9DQ, U.K. www.research-instruments.com

20. Sanderson A.C. and Weiss L.E., 1982, "Image-based Visual Servo Control of Robots," *26th Annual SPIE Technical Symposium.*

21. Weiss L.E., Sanderson A.C., and Neuman C.P., 1987, "Dynamic Sensor Based Control of Robots with Visual Feedback," IEEE J. Robotics and Automation, Vol. R.A-3, No. 5, pp. 404-417, October.

22. Wim de Leeuw W. and Robert van Liere R. "Chromatin Decondensation: A Case Study of Tracking Features in Confocal Data," Center for Mathematics and Computer Science, CWI, Amsterdam, Netherlands.

23. Xudong L., Zong G. and Bi S., 2001, "Development of Global Vision System for Biological Automatic Micro-Manipulation System" *Proceedings of the 2001 IEEE International Conference on Robotics and Automation,* pp 127-132, May, Seuol, Korea.

24. Yu S. and Nelson B. J. 2003, "Autonomous Injection of Biological Cells Using Visual Servoing", Advanced Microsystems Lab, Department of Mechanical Engineering, University of Minnesota, Twin Cities Minneapolis, MN 55455.

25. Young D., Glasbey C.A., Gray A.J. and Martin N.J., 1995, "Identification and Sizing of Cells in Microscope Images by Template Matching Method and Edge Detection", *IEE Conference on Image Processing and its Applications,* No. 410, pp 266-270, July.

26. Zhou Y., Nelson B.J., and Vikramaditya B., 1998, "Fusing Force and Vision Feedback For Micromanipulation", *Proceedings of the IEE ICRA,* pp. 1220-1225, May, Leuven.

27. Horgan G.W., Reid C.A., and Glasbey C.A., 2000, "Biological Image Processing and Enhancement", pp 37 – 67 , *Image Processing and Analysis – A Practical Approach,* Oxford University Press Inc., New York, USA.